AD-A061 771 PHOENIX CORP MCLEAN VA F/G 17/7

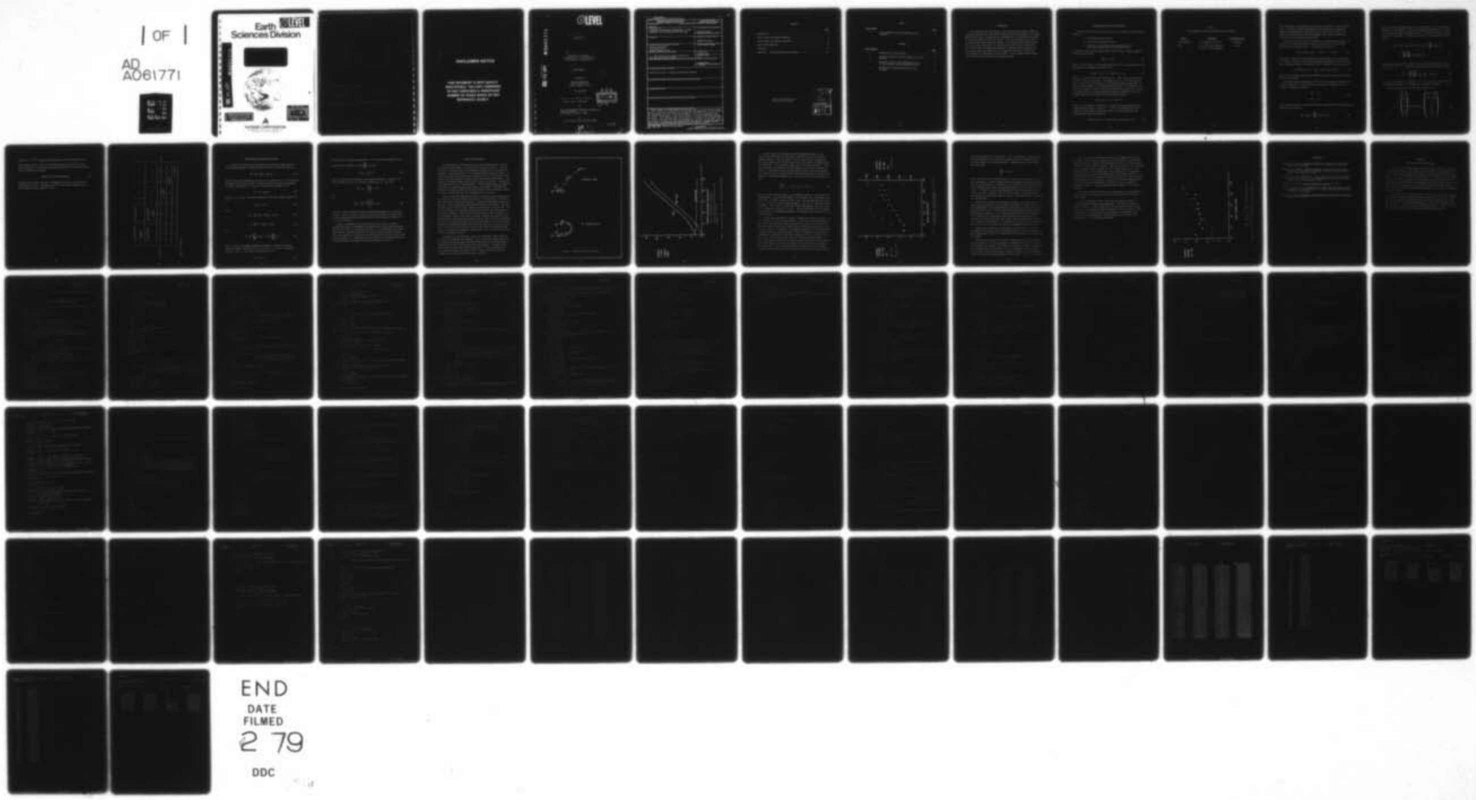
OPTIMIZED, POST-MISSION DETERMINATION OF THE DEFLECTION OF THE --ETC(U)

OCT 78 DAAK70-78-C-0069

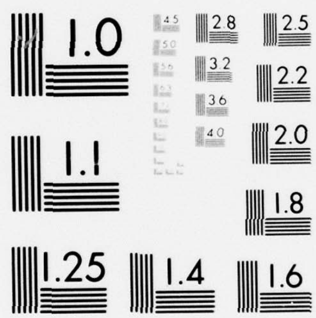
UNCLASSIFIED ETL-0164 NL

1 OF 1

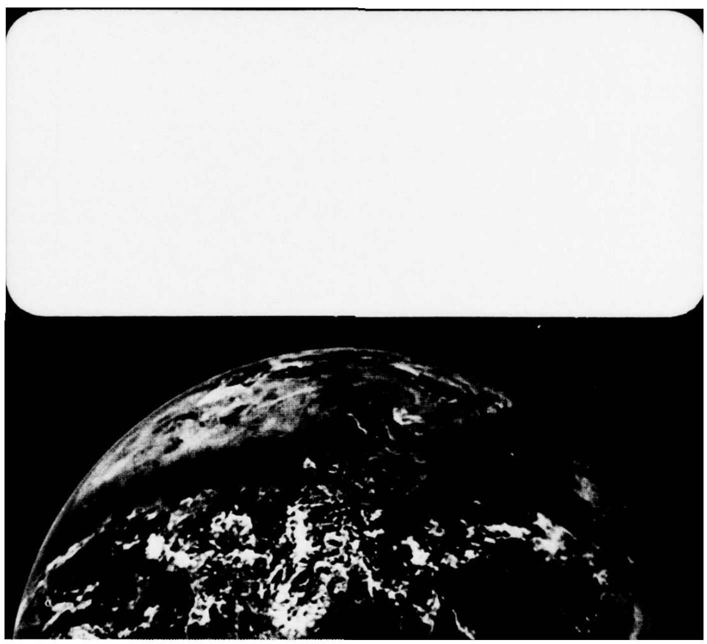
AD
A061771



END
DATE
FILMED
2 79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



DISCLAIMER NOTICE

LEVEL II

AD A061771

ETL-0164

Optimized, Post-Mission Determination of the Deflection of the Vertical Using RGSS Data

Final Report

Prepared by

Phoenix Corporation
1600 Anderson Road
McLean, Virginia 22102

October 1978

for

U.S. Army Engineer Topographic Laboratories
Research Institute
Fort Belvoir, Virginia 22060

Under Contract No. DAAK70-78-C-0069

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0164	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) OPTIMIZED, POST-MISSION DETERMINATION OF THE DEFLECTION OF THE VERTICAL USING RGSS DATA		5. TYPE OF REPORT & PERIOD COVERED Contract Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DAAK70-78-C-0069 New
9. PERFORMING ORGANIZATION NAME AND ADDRESS Phoenix Corporation 1600 Anderson Road McLean, Virginia 22102		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		12. REPORT DATE October 1978
		13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is a continuation of an earlier report on a potentially optimal method of recovering deflections of the vertical from RGSS data. In this report, the implementation of the method and estimates of the errors associated with the method are described. In the first section, an optimal weighting technique is derived. This technique also leads directly to a priori error estimates. Next, the results from using the method on hypothetical traverses are described. From these data, it appears that the optimal method can lead to a significant reduction in the errors in estimating the deflections of the vertical. A final appendix gives instructions for the use of the associated computer program.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

Appendix Not Necessary For
Understanding Of Report

TABLE

FIGURES

INTRODUCTION

In an earlier report (Lyon et al., 1977), a potentially optimal method of recovering deflections of the vertical from RGSS data was described. This report continues that work — describing the implementation of the method and estimates of the errors associated with the method. In the first section of the report an optimal weighting technique is derived. This technique also leads directly to a priori error estimates. The second section describes the results from use of the method on hypothetical traverses. From these data it appears that the optimal method can indeed lead to a significant reduction in the errors in estimating the deflections of the vertical. A final appendix gives instructions for the use of the associated computer program.

ERROR SOURCES AND OPTIMAL WEIGHTING

There are three sources of error which will be considered in this report. These are

1) correlated gyroscope errors

2) correlated accelerometer errors

3) errors due to the colocation determination of the deflections during each leg - colocation error.

We assume that the gyro and accelerometer errors follow a Langevin equation (first order Markovian)

$$\frac{d\alpha}{dt} + \nu\alpha = A(t) \tag{1}$$

where ν is the inverse of the correlation time and $A(t)$ is Gaussian white noise. This leads to a covariance (Papoulis, 1965)

$$< \alpha(t_1)\ \alpha(t_2) > = \alpha_o^2 \exp(-\nu|t_2 - t_1|) \tag{2}$$

where α_o^2 is $< \alpha(t)\, \alpha(t) >$, the variance of α. Table 1 gives the values for α_o^2 and ν used for this report. The assumption of eq. (1) is not strictly true. In particular, the purpose of the reduction scheme outlined here and in the previous report is to estimate gyro drift rates. The deviations, $\delta\alpha$, from this estimate, $\bar{\alpha}$, say, will not be distributed as equation (1). However, it is clear that if the total covariance is given by equation (2), then the maximum value that the variance about the mean may attain is

$$\mathrm{Var}\ (\alpha - \bar{\alpha}) \leq \alpha_o^2 (1 - \exp(-\nu T))$$

where T is the length of the mission. Similarly, the correlation time for values about the mean, τ, must be $\tau \leq T$. Thus, even though equation (1) does not strictly apply to variations about $\bar{\alpha}$, we will assume that that variation holds and the $\delta\alpha$(gyro) are

$$< \delta\alpha(t_1)\ \delta\alpha(t_2) > = \alpha_o^2 (1 - \exp(-\nu T)\ \exp(-1t_1 - t_2|/T) \tag{2a}$$

Table 1

Error Parameters Used for Deflection Error Estimates

Source	RMS Value	Correlation Time
Accelerometers	10 microg's (all axes)	40 minutes
Gyros	2.5×10^{-3} °/hr (horizontal 2.0×10^{-3} °/hr (vertical)	2 hours

The accelerometers have appreciable white noise in addition to the correlated noise. This may be handled approximately by increasing the accelerometer variances and decreasing the correlation time. The values in Table 1 are based on the data given by Huddle and Maughmer (1972) suitably modified in accordance with the discussion given above.

The colocation errors may be estimated in a straightforward fashion. We assume that the actual deflection covariance function is the second order Markovian given by Kasper (1971). The colocation variance is

$$<(r_i - r_i^e)(r_j - r_m^e)> \;=\; <r_i r_j> \;-\; <r_i^e r_j^e> \tag{3}$$

where the r notation for the deflections was introduced in the first report. r_j is either a north or east deflection depending on whether j is odd or even. The e superscript denotes the estimated value. Equation (3) may be reduced to

$$<(r_i - r_i^e)(r_j - r_j^e)> = <r_i r_j> - <r_i \tilde{r}_\kappa><r_j \tilde{r}_\ell><\tilde{r}_\ell \tilde{r}_\kappa>^{-1} \tag{4}$$

where the tilde denotes deflections belonging to the basis set from which the others are estimated.

The basic data available relate to u^n and v^n, the north and east velocity errors, respectively, at the end of the n-th leg of the mission. For convenience, introduce the notation

$$\begin{aligned} W_1{}^n &= u^n \\ W_2{}^n &= v^n \end{aligned} \tag{5}$$

Then the basic equation for the error velocities, equation (40) of the original report can be written as

$$W_\ell^n - A_{\ell\kappa}^n \mu_\kappa^o + \sum_{j=0}^{n-1} B_{\ell\kappa j}^n \psi_\kappa^j \equiv F_\ell^n \tag{6}$$

where A and B are matrices defined in the first report. μ_κ^o are the initial conditions of the solution. ψ_κ^j is the inhomogeneous driving term. If there are no errors F_ℓ^n should be identically zero. If errors are present F_ℓ^n will, in general, be non-zero. To estimate the errors, we square equation (6) to obtain

$$(F_\ell^n)^2 = (W_\ell^n - A_{\ell\kappa}^n \mu_\kappa^o)^2 + 2(W_\ell^n - A_{\ell\kappa}^n \mu_\kappa^o) \sum_{j=o}^{n-1} B_{\ell\kappa j}^n \psi_\kappa^j$$

$$+ \sum_{j=0}^{n-1} \sum_{m=0}^{n-1} B_{\ell\kappa j}^n B_{\ell o m}^n \psi_\kappa^j \psi_o^m \tag{7}$$

We assume no errors in $(W_\ell^n - A_{\ell\kappa}^n \mu_\kappa^o)$ and that the expectation value of the errors in ψ_κ^n, $\langle\psi_\kappa^n\rangle = 0$. Taking the expectation value of equation (7) gives

$$\langle F_\omega^{n^2} \rangle \sum_{j=0}^{n-1} \sum_{m=0}^{n-1} B_{\ell\kappa j}^n B_{\ell o m}^n \langle \delta\psi_\kappa^j \delta\psi_o^m \rangle \tag{8}$$

where $\delta\psi_\kappa^j = \psi_\kappa^j$ (assumed) $- \psi_\kappa^j$ (true). $\langle F^{n^2} \rangle$ is, of course, the variance of the observed data point. It remains only to evaluate $\langle \delta\psi_\kappa^j \delta\psi_o^m \rangle$.

We use the ordering of ψ of the first report, i.e.,

$$\psi^n = \begin{pmatrix} -g\xi^n \\ g\eta^n \\ 0 \\ \alpha^n \\ \beta^n \\ \gamma^n \end{pmatrix} \quad \text{and, then } \delta\psi^n = \begin{pmatrix} -g\delta\xi^n + \delta a_N^n \\ g\delta\eta^n + \delta a_E^n \\ 0 \\ \delta\alpha^n \\ \delta\beta^n \\ \delta\gamma^n \end{pmatrix} \tag{9}$$

with $\delta\xi^n = \xi^n - \xi^{n(e)}$, δa_N^n and δa_E^n the north and east accelerometer errors, respectively, and $\delta\alpha^n$, $\delta\beta^n$, $\delta\gamma^n$ the correlated gyro errors for Z, N, and E axes, respectively. Neglecting zero cross-correlations, the error covariance matrix in equation (8) becomes

(Equation 10 on following page) (10)

Equations (8) and (10) then give an estimate of the errors associated with the measurement of W_ℓ^n. Furthermore $1/\langle F_\ell^{n^2} \rangle$ is the optimal weighting for the least squares solution (Brownlee, 1962).

$$
\langle \delta\psi^j \delta\psi^m \rangle = \left|
\begin{array}{c:c:c:c:c:c}
\begin{matrix} g^2 \langle \delta r^{2j-1} \delta r^{2m-1} \rangle \\ + \langle \delta a_N^j \delta a_N^m \rangle \end{matrix} & - g^2 \langle \delta r^{2j-1} \delta r^{2m} \rangle & 0 & 0 & 0 & 0 \\
\hdashline
- g^2 \langle \delta r^{2j} \delta r^{2m-1} \rangle & \begin{matrix} g^2 \langle \delta r^{2j} \delta r^{2m} \rangle \\ + \langle \delta a_E^j \delta a_E^m \rangle \end{matrix} & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & \langle \delta\alpha^j \delta\alpha^m \rangle & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & \langle \delta\beta^j \delta\beta^m \rangle & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & \langle \delta\gamma^j \delta\gamma^m \rangle \\
\hdashline
\end{array}
\right| \qquad (10)
$$

where $\delta r^j = r^{j(e)} - r^j$

ERROR ESTIMATE FOR DERIVED QUANTITIES

We need now to derive error estimates for the derived quantities in the least squares solution. We rewrite equation (42) of the original report as

$$F^n_\ell = W^n_\ell - A^n_{\ell\kappa}\mu^o_\kappa - D^n_{\ell m}\, h_m \tag{11}$$

where the h_m are the quantities in which we are interested, i.e., the calculated deflections and gyro drift rates. Change the notation slightly by replacing the double index (n,ℓ) by $j = 2(n-1) + \ell$ and rewrite equation (11) as

$$F_j = a_j - D_{jm}\, hm \tag{12}$$

where $a_j = W_j - A_{j\kappa}\mu^o_\kappa$. The matrix equation for the least squares solution for h is

$$E_{ij}\, h_j - b_i = 0 \tag{13}$$

where

$$E_{ij} = \sum_\kappa (D_{\kappa i} - \bar{D}_i)(D_{\kappa j} - \bar{D}_j)\, W_\kappa \tag{14}$$

$$b_i = \sum_\kappa (a_\kappa - \bar{a})(D_{\kappa i} - \bar{D}_i) W_\kappa \tag{15}$$

and

$$\bar{D}_i = \frac{1}{N} \sum_{\kappa=1}^{N} D_{\kappa i} W_{\kappa j} \qquad \bar{a} = \frac{1}{N} \sum_{\kappa=1}^{N} a_\kappa W_\kappa \tag{16}$$

with $W_\kappa = 1/\sigma_\kappa^2$, the optimal weighting discussed in the previous section. Writing the normal equation (13) as we have leads to a number of advantages (Brownlee, 1962). The solution of equation (13) is

$$h_j = E_{ji}^{-1} b_i \tag{17}$$

The inverse $\underset{\sim}{E}^{-1}$ has special properties. If $\bar{\sigma}^2$ is the mean variance of the observed error velocity, i.e., $\frac{1}{N} \sum_{\kappa=1}^{N} \sigma_\kappa^{\ 2}$, then

$$\mathrm{Var}\ h_j = E_{jj}^{-1} \bar{\sigma}^2 \tag{18}$$

Thus, we have an error estimate of the derived quantities. Further, if we wish to throw out one of the solved for quantities, h_μ, say, then

$$h_j' = h_j - \frac{E_{j\mu}^{-1} h_\mu}{E_{\mu\mu}^{-1}} \quad j \neq \mu \tag{19}$$

and

$$E_{j\kappa}^{-1\,'} = E_{j\kappa}^{-1} - \frac{E_{j\mu}^{-1}\ E_{\kappa\mu}^{-1}}{E_{\mu\mu}^{-1}} \quad j,\kappa \neq \mu \tag{20}$$

where the ' denotes quantities where the assumed dependence on $h\mu$ has been removed. Thus, if we wish to remove the gyro drift rates, for example, from the least squares solution and see how much the deflections are affectd, it can be done trivially.

As will be discussed in the results section, we follow a somewhat different procedure in eliminating variables from the least squares solution. Equation (20) holds if no weighting is used, or if the weighting is unchanged after removing a variable from the fit. Since we remove variables that do affect the weighting, we use the more laborious method of starting from scratch with new weights. It is important to note, however, that equation (18) still holds and provides an estimate of the errors involved in the fit.

RESULTS AND DISCUSSION

Two hypothetical traverses were used to find estimated errors for the outlined reduction method. The courses are sketched in Figure 1. The first traverse is a straight line to the northeast covering 25 km. The second is polygonal - also covering 25 km. The assumed vehicle speed was 25 km/hr - so that total travel time was one hour, not counting stops. Deflections of the vertical were determined at 10 points evenly spaced along the traverses. The vehicle was assumed to stop either 20 or 40 times on a mission. This made the least squares system well overdetermined. It also helped produce an answer to the question of whether fewer or more stops is preferable. Solutions were generated for cases including all the gyro drifts as fitted variables, including just the horizontal axes, and including none of the gyros.

Figure 2 shows the estimated variance in the north velocity channel over the course of a 20 stop straight line traverse. There are only two significant contributors to the total variance - the north accelerometer and the east axis gyro. Their contributions are plotted separately in Figure 2. The accelerometer error is more or less constant over the course of the mission. The gyro drift becomes the dominant contribution early on in the traverse and is constantly increasing. The value for the gyro variance used for Figure 2 is that assuming a constant drift rate is removed. Without the removal of the average drift, the gyro-related variance would be about a factor of two bigger. The form of the equivalent curves for the 40 stop traverse is nearly identical. However, the individual variances are about half what they are for the 20 stop case. This is just a reflection of the fact the errors at individual stops appear to accumulate as individual random events. Half the time then implies half the accumulated variance.

The results for the polygonal course are, once again, nearly identical to that for the straight line traverse. This is a direct consequence of the fact that the estimated errors introduced from collocation are negligible in comparison with those from the gyros and accelerometers. According to the model used here these significant error sources are almost independent of the direction in which the vehicle travels. Hence, the results from the polygonal and straight traverses are almost identical.

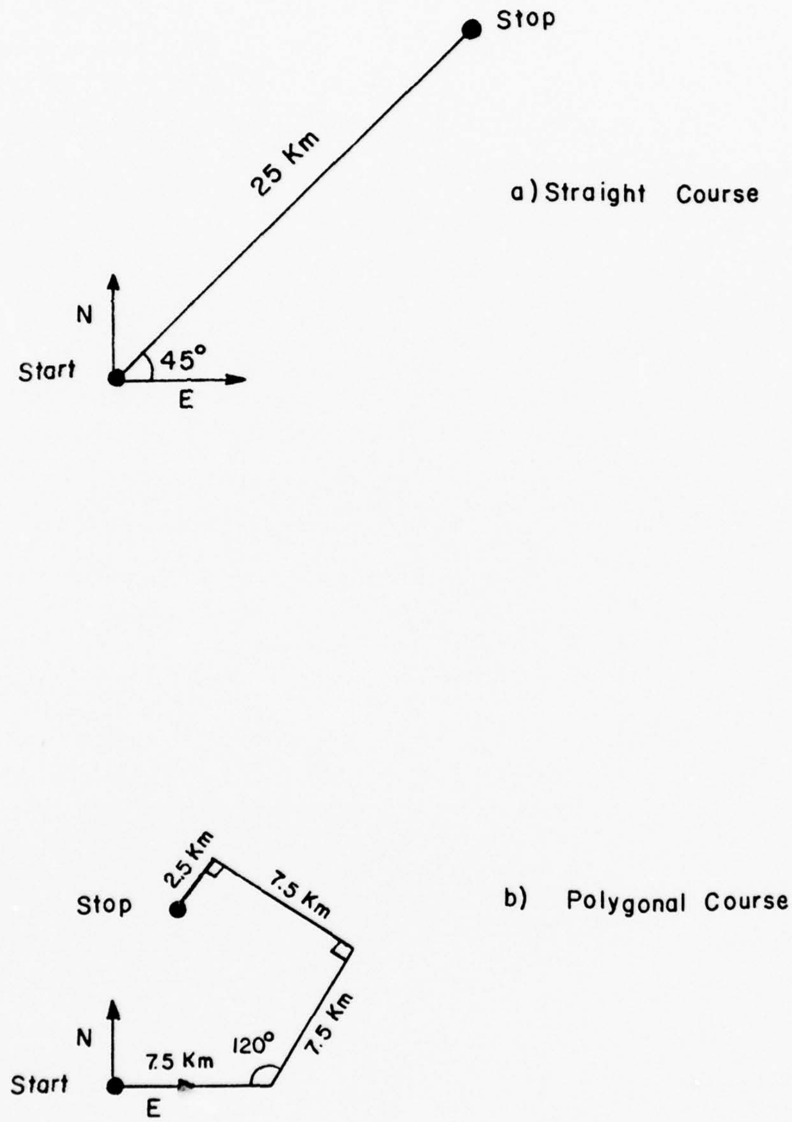
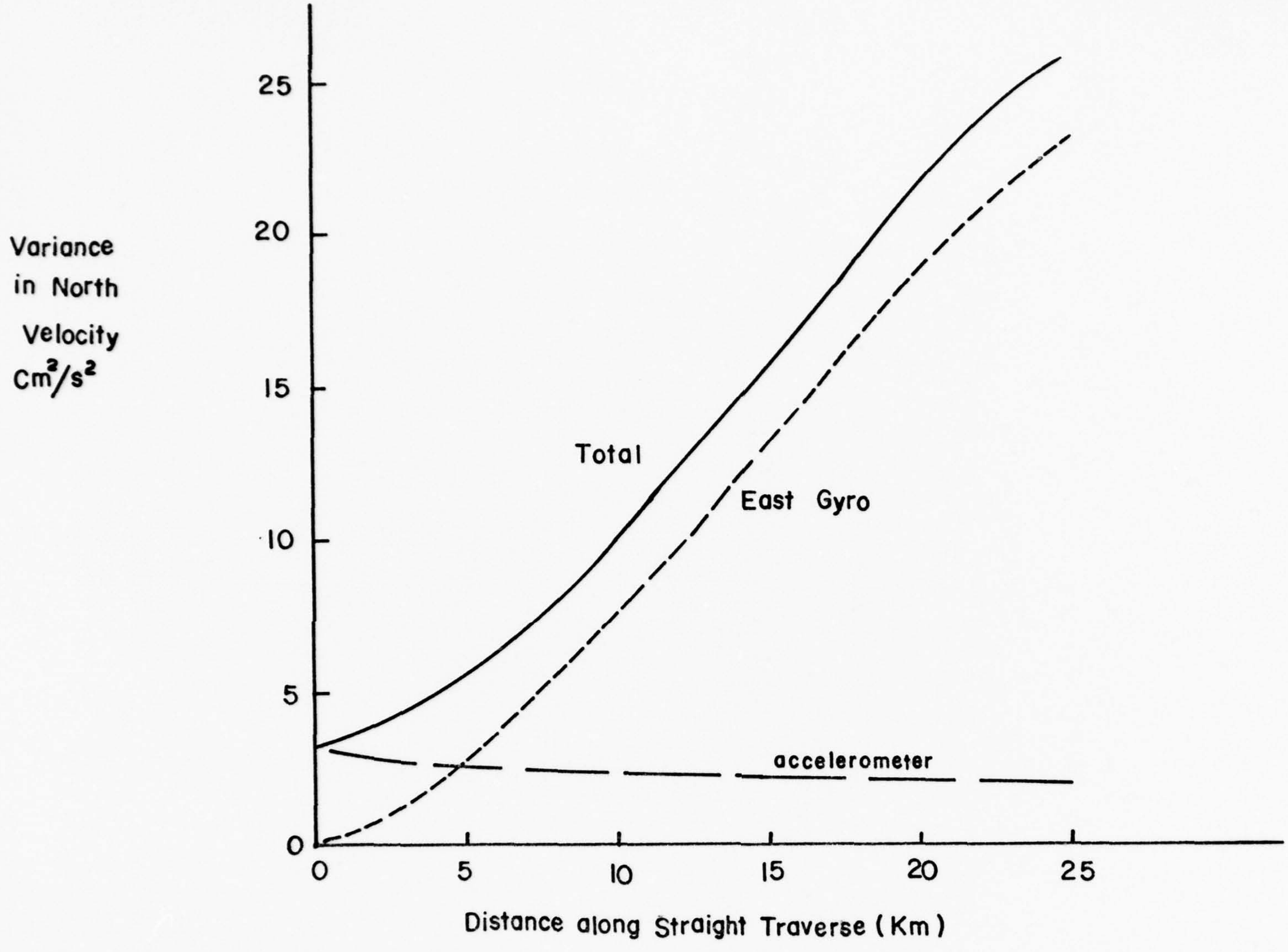


Figure 1. Hypothetical Traverse Courses



---- Variance due to Gyro drift

— — — Variance due to accelerometer noise

——— Total variance

Figure 2. Estimated Variances in North Channel Over 25 km Traverse

Figure 3 shows the estimated errors (standard deviation) for the derived north deflection of the vertical using 20 stops and an optimally weighted solution. The first point to note is the terrible performance of the fitted solution when all three gyro drifts are included. Maximum errors are almost 200" . This behavior occurs only in the north direction. In the east direction, the solution is as well behaved as the other two curves in Figure 3. The reason for this strange behavior can be found by considering the equations for the east gyro error, the vertical gyro error, and the error velocity. Taking those equations (7, 10, and 12 from the first report), we find

$$\frac{d^2\phi_E}{dt^2} = - r_s^2 \phi_E + r_s^2 \xi - r \cos \phi \alpha + . . . \tag{21}$$

where ϕ_E is the east gyro error, r_s the Schuler frequency, r the terrestrial rotation rate, ξ the north deflection, ϕ the latitude, and α the vertical gyro drift rate. The point to note is that ξ and α come into equation (21) in the same way. Thus, in a least squares solution, ξ and α are to some extent interchangeable. Since there is no vertical channel information, there is no real way to separate the effects of α from ξ. The east deflection is well behaved because there is no comparable coupling of the vertical gyro drift rate to the east deflection.

A significant improvement is made by removing the vertical gyro drift from the solution, as can be seen from Figure 3. The results are still somewhat puzzling as the estimated errors are virtually the same in the case where the horizontal gyro drift rates are included in the solution as when they are not. This is in spite of the fact that the assumed variances of the error velocities is about a factor two smaller when gyro drifts are included in the solution. Unfortunately, inspection of the error covariance matrix, eq. (18-20), shows that ξ and γ (the east gyro drift rate) are strongly anti-correlated, i.e. have a large negative covariance in the structure of the least square solution. This implies that the situation is similar to that discussed with respect to the vertical gyro. That is, with the given information, the least squares solution has difficulty telling the difference between an east gyro drift rate

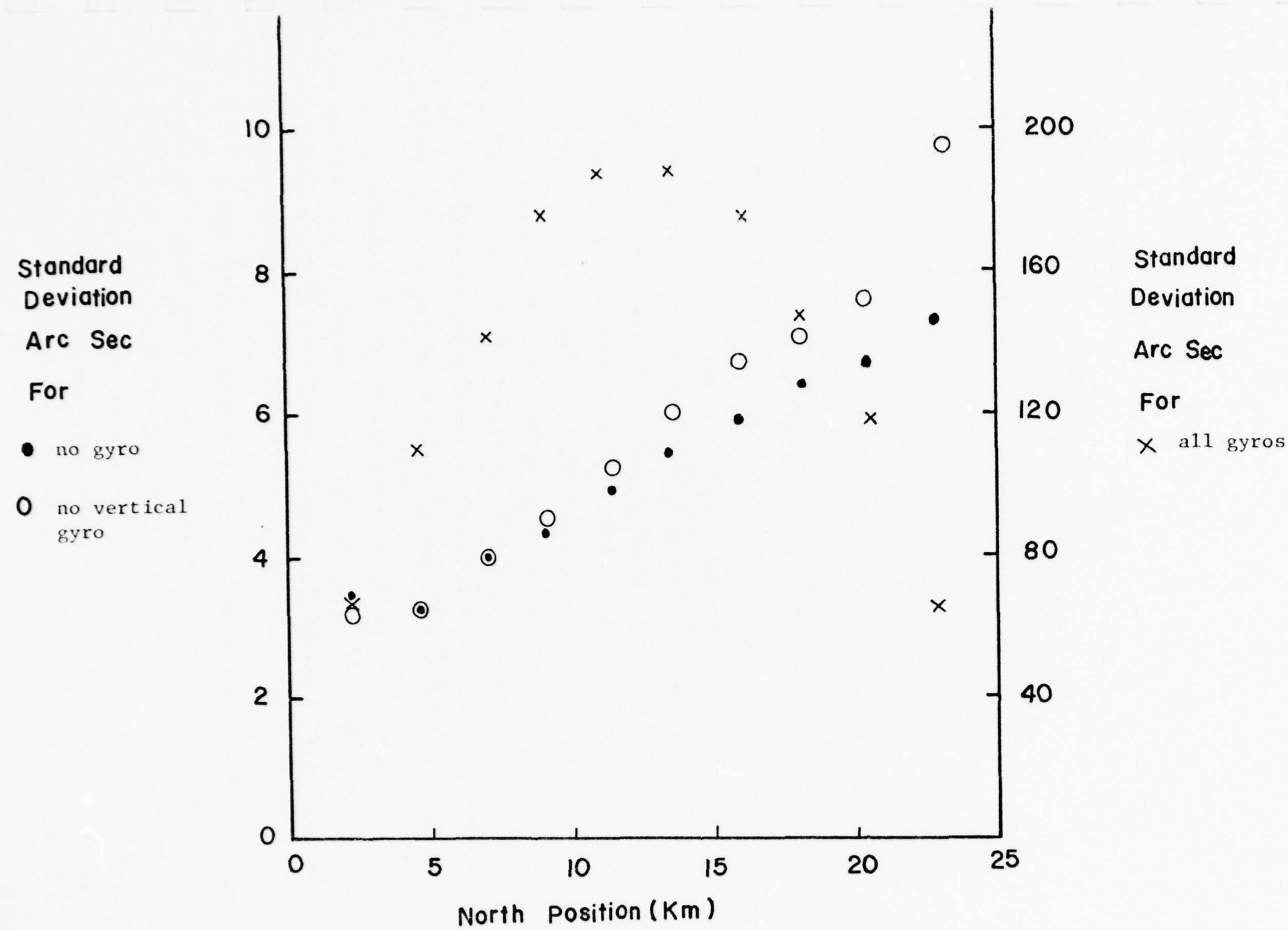


Figure 3. Estimated Errors in North Deflection (ξ) for 20 Step Straight Traverse Weighted Solution

and a north deflection of the vertical. This, on reflection, should not be terribly surprising. We can write an equation similar to equation (21) for the error velocity in the north direction. Keeping just the largest terms, this looks like

$$\frac{d^2u}{dt^2} = - r_s^2 u + \gamma \qquad (22)$$

where u is the north error velocity and γ is the east gyro drift rate. The north deflection, ξ, enters equation (22) only through the boundary conditions. Another way of looking at the problem is that equation (22) describes a sinusoidal variation whose phase and amplitude depends on the relative sizes of ξ and γ. In effect, the phase and amplitude must both be determined by one number - the error velocity. What is needed to help is information about, for example, the acceleration error at a stop. This would serve to disentangle the two quantities.

Actually, the situation is not nearly so bleak as has been painted. The values for the variation of the gyro drift rate about the mission mean are quite conservative. The actual variance could easily be a factor of two lower than what we have used. In this case the solution including the gyro rates would clearly be superior. It is interesting to note that at the beginning of the mission when accelerometer errors dominate the variance, the two solutions are almost identical. This implies that the accelerometer errors give a limit to the accuracy of the recovery of the deflection in the neighborhood of 2-3" for the 20 stop case and the accelerometer parameters used.

Somewhat better results are obtained by using 40 stops. The gain is essentially by the square root of the number of stops. Thus, a 40 stop case gives errors about a factor $\sqrt{2}$ better than the 20 stop case, all other things being equal.

Since the major sources of error have essentially just a time dependence and not a position or velocity dependence, speeding up the rate of traverse also increases the accuracy. The increase in accuracy is roughly proportional to the square root of the velocity. This, of course, has a limit when the vehicle velocity becomes high enough to make the neglect of velocity dependent terms in the error propagation equations (first report: eq. (1) - (6)) serious.

So far the results discussed have dealt with weighted least squares solutions. Figure 4 shows a comparison of error estimates for the derived north deflections. Inherent in an unweighted solution is a single assumed variance for all the error velocities. This is in contrast to the increasing - as a function of time - variances in the weighted solution. It is not surprising, then, that the deflection error estimates for the unweighted solution tend to be more uniform than those of the weighted solution. If the error model used is reasonable then the errors derived from the weighted solution should be more accurate. In practice, the derived deflections do not seem to be greatly affected by the choice of either a weighted or unweighted solution. Thus, the weighted solution appears to be slightly preferable.

The results presented in Figure 3 are comparable to those presented by Huddle (1973) in his discussion of the Position and Azimuth Determining System (PADS).

We have argued above that a factor two improvement on Figure 3 is easily attainable without system improvement. This would be superior to the PADS results. If more information can be obtained from the inertial system - i.e., acceleration errors in addition to velocity errors at each of the stops - the accuracy of the system should be determined by the accuracy of the accelerometers, and deflections of the vertical with accuracies of 1 - 1.5" should be attainable.

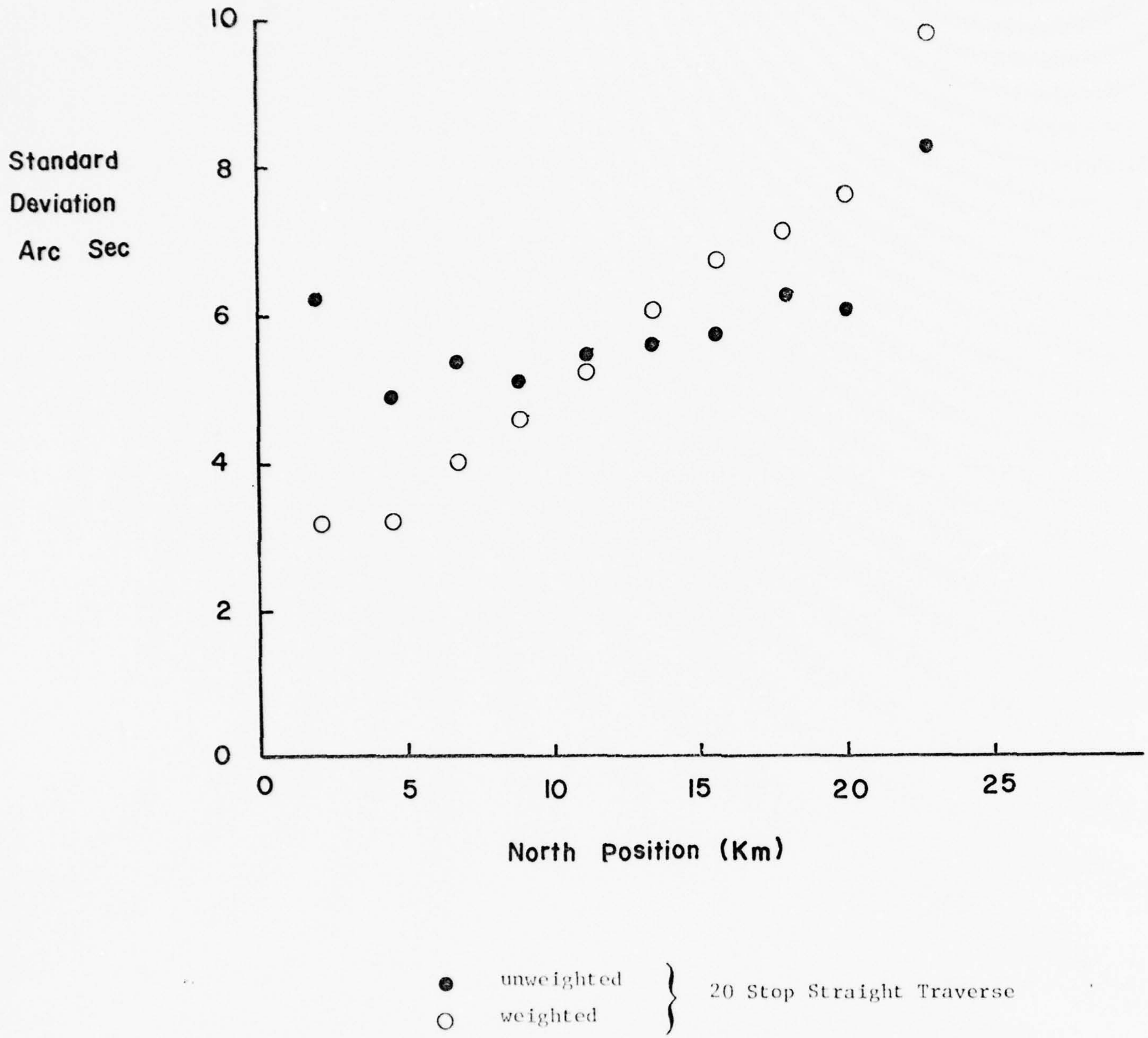


Figure 4. Differences in North Deflection (ξ) Error Estimates

REFERENCES

Brownlee, R. B., 1962, <u>Statistical Methods for Scientists and Engineers</u>, Prentice Hall, New York.

Huddle, J. R., 1973, "A Study and Analysis of the Position and Azimuth Determining System (PADS) for Mapping, Charting, and Geodesy Applications", Litton Report No. 402215.

Huddle, J. R. and Maughmer, R. W., 1972, "The Application of Error Control Techniques in the Design of an Advanced Augmented Inertial Surveying System", Litton Publication No. 11678 A.

Kasper, J. F., 1971, <u>Journal of Geophysical Research</u>, <u>76</u>, 7844.

Lyon, J., Mader, G. L., and Heuring, F. T., 1977, "Optimized Method for the Derivation of the Deflection of the Vertical from RGSS Data", Phoenix Corporation, Final Report.

Papoulis, A., 1965, <u>Probability, Random Variables, and Stochastic Processes</u>, McGraw-Hill, New York.

APPENDIX 1

THE OPTIMIZED REDUCTION PROGRAM

A listing of the FORTRAN Program used to determine deflections, and error estimates is given below. The input data are described in the comment cards at the beginning of the program and should be self-explanatory with two exceptions: 1) all input data are cgs and angles are in radians, and 2) the program is set up to handle a traverse with known deflections at the start and stop. To use only known deflections at the beginning three things must be done. First, add a dummy finishing stop to the data with the position of the finish equal to the start. Second, set XIFIN and ETAFIN equal to XIϕ and ETAϕ, respectively. Third, set IDEFL = 1.

The output format is also shown below. Solutions are given for cases with all gyro rates, horizontal gyro rates only, and no gyro rates in turn. Before the actual solution the estimated variances of the error velocities is given both in total and from the individual sources. The solved-for quantities are each presented with error estimates (standard deviations). Finally, for each case, the actual deviation of the solution from the data is given.

```
      PROGRAM GUIDE(INPUT,OUTPUT,DEBUG=OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C*    ARRAYS
C*    CALLS
C
C         PROGRAM FOR THE OPTIMIZED, POST-MISSION DETERMINATION OF
C         THE DEFLECTION OF THE VERTICAL USING RGSS DATA
C                 PROGRAM PRODUCED BY PHOENIX CORPORATION
C
C  ************************************************************************
      COMMON CPHI,SPHI,TPHI
      COMMON/SCRTCH/COVAC,COGY
      DOUBLE PRECISION PSI,XL
      DOUBLE PRECISION DUMMY,DUM2,DUM3,DUM4
      DOUBLE PRECISION SOMAT,SUMAT
      DIMENSION K3(50)
      DIMENSION TGO(51),TSTOP(51),U(51),V(50),XX(51),YY(51),XHAT(51),
     2 YHAT(51),X(50),Y(50),START(6),STEM(6)
      DIMENSION PSI(50,6,6),XL(50,6,6),DUMMY(6,6),DUM2(6,6),DUM3(6,6),
     2DUM4(6,6)
      DIMENSION COEF(80,24),DEFL(80,80),COVAR(80,24)    ,
     1 SOMAT(23,24),F(100),VAR(100),SUMAT(24)
      DIMENSION COMEAN(24),CONT (80,80),TIME(41) ,COGY(3,40,40)
      DIMENSION VARGY(3) ,TAUGY(3),GYVAR(3,80,40),GYRO(3,80),VARCO(80)
      DIMENSION ACVAR(80),WATE(80),COVAC(2,40,40),ACSIG(2),ACTAU(2)
      DIMENSION TARGY(3),TAUTY(3)
      DATA G/9.80665E2/
      SGIN(I,J) = 1. - 2.* AMOD(FLOAT(I+J),2.)
C
C         INPUT DATA
C             N1 = # OF POINTS AT WHICH A POSITION IS  SPECIFIED
C                 THE FIRST AND LAST POINTS BEING THE START AND STOP
C                 OF THE VEHICLE - THE REMAINDER THE POINTS AT WHICH THE
C                 DEFLECTION IS TO BE DETERMINED.
C            MSTOP = NUMBER OF VEHICLE STOPS IN A GIVEN MISSION
C            TGO(I) = TIME SPENT TRAVELLING ON I-TH LEG
C            TSTOP(I) = TIME SPENT STOPPED ON I-TH LEG
C            U(I) = X OR NRTH VELOCITY ERROR AT END OF I-TH LEG
C            V(I) = EAST OR Y VELOCITY ERROR AT END OF I-TH LEG
      READ(5,1)MSTOP,N1
    1 FORMAT(2I5)
      N2 = 2 * N1
      M2 = 2 * MSTOP
      MS = MSTOP + 1
      READ(5,2)(K3(I),TGO(I),TSTOP(I),U(I),V(I),XX(I),YY(I),I=1,MS)
    2 FORMAT(I10,6F10.4)
      WRITE(6,101)(I,TGO(I),TSTOP(I),U(I),V(I),XX(I),YY(I),I=1,MS)
  101 FORMAT(1X,I5,6E15.4)
      READ(5,9)(K9(I),XHAT(I),YHAT(I),I=1,N1)
    9 FORMAT(I10,2F10.4)
      WRITE(6,102)(K9(I),XHAT(I),YHAT(I),I=1,N1)
  102 FORMAT(1H1,(I5,2E14.4))
      READ(5,3)FLAT,XI0,ETA0,XIFIN,ETAFIN
      WRITE(6,3) FLAT,XI0,ETA0,XIFIN,ETAFIN
```

```
      READ(5,3)(START(I),I=1,6)
      WRITE(6,7)(START(I),I=1,6)
      READ(5,2)IWATE
      WRITE(6,101) IWATE
    3 FORMAT(6E12.4)
      READ(5,3) (ACSIG(I),ACTAV(I),I=1,2)
      WRITE(6,3)(ACSIG(I),ACTAV(I),I=1,2)
      READ(5,3) (TAPSY(I),TAUTY(I),I=1,3)
      WRITE(6,3)(TAPGY(I),TAUTY(I),I=1,3)
C     CPHI =DCOS(TLAT)
      CPHI = COS(TLAT)
C     SPHI =DSIN(TLAT)
      SPHI = SIN(TLAT)
      TPHI = SPHI/CPHI
      CALL ADVANS(0.,0.,PSI,XL,0,X,Y)
      DO 150 I=1,N2
      DO 150 J =1,N2
  150 DEFL(I,J) = 0.
      DO 200 I = 1,MSTOP
         X(I) = (XX(I+1) + XX(I))/2.
  200    Y(I) = (YY(I+1) + YY(I))/2.
      DO 300 I = 1,MSTOP
      CALL TIMEX(TGO(I),TSTOP(I),PSI,XL,I,XX,YY)
  300 CONTINUE
      DO 1000 I=1,MSTOP
      CALL OEQUIV(I,PSI,DUMMY)
      DO 1010 KK = 1,6
      STEM(KK) = 0.
      DO 1010 L = 1,6
 1010 STEM(KK) = STEM(KK) + DUMMY(KK,L) * START(L)
      DO 1020 L = 1,6
 1020 START(L) = STEM(L)
      COEF(2*I-1,N2) = U(I) - START(1)
      COEF(2*I,N2) = V(I) - START(2)
 1000 CONTINUE
C
C
C         DETERMINATION OF THE TERMS IN THE COEF MATRIX WHICH DEPEND
C         ON THE  DEFLECTIONS OF THE VERTICAL
C
C         FIRST STEP' DEFINE A MATRIX - DEFL - WHICH GIVES THE VARA
C         FIRST STEP' DEFINE A MATRIX - DEFL - WHICH GIVES THE DEPENDENCE
C                 OF THE VELOCITY ERRORS ON THE VALUES OF THE
C                 DEFLECTIONS AT EACH OF THE MIDPOINTS OF EACH TRAVEL LR
C                 DEFLECTIONS AT EACH OF THE MIDPOINTS OF EACH TRAVEL LEG
C
      DO 1100 I = 1,MSTOP
      CALL DEQUIV(I,XL,DUMMY)
      DEFL(2*I-1,2*I-1) = DUMMY(1,1)
      DEFL(2*I-1,2*I) = DUMMY(1,2)
      DEFL(2*I,2*I-1) = DUMMY(2,1)
      DEFL(2*I,2*I) = DUMMY(2,2)
      IF (I.EQ.MSTOP) GO TO 1100
```

```
      KK = I + 1
      DO 1110 N = KK,MSTOP
      CALL DEQUIV(N,PST,DUM2)
      CALL DXMAT(DUM2,DUMMY,DUM3,6,6,6)
      CALL DEQUAL(DUM3,DUMMY)
      DEFL(2*N-1,2*I-1) = DUMMY(1,1)
      DEFL(2*N-1,2*I) = DUMMY(1,2)
      DEFL(2*N,2*I-1) = DUMMY(2,1)
      DEFL(2*N,2*I) = DUMMY(2,2)
 1110 CONTINUE
 1100 CONTINUE
C
C         SECOND STEP* DEFINE THE MATRIX - COVAR - WHICH , WHEN
C                 MULTIPLIED BY DEFL GIVES THE DEPENDENCE OF THE COEF
C                 MATRIX ON THE DEFLECTIONS AT THE DESIRED POINTS
C
      CALL COLLOC(MSTOP,N1,COVAR,X,Y,XHAT,YHAT,COWT)
      N2P = N2 - 4
C
C         THIRD STEP* DO THE MULTIPLICATION
C
      DO 1150 I = 1,N2
      DO 1150 J = 1,N2P
      COEF(I,J) = 0.
      DO 1150 L = 1,N2
 1150 COEF(I,J) = COEF(I,J) + DEFL(I,L) * COVAR(L,J+2)
      WRITE(6,10) COEF
      DO 1160 I = 1,N2
C
C         UPDATE THE CONSTANT PART OF THE COEF MATRIX  TO ACCOUNT FOR THE
C                 KNOWN VALUES OF THE DEFLECTION AT THE START AND STOP
C                 OF THE MISSION
C
      DO 1155 L = 1,N2
      COEF(I,N2) = COEF(I,N2) - G * XI0 * DEFL(I,L) * COVAR(L,1)
     2                        - G * ETA0* DEFL(I,L) * COVAR(L,2)
     3                        - G *XIFIN* DEFL(I,L) * COVAR(L,N2-1)
     4                        - G *ETAFIN*DEFL(I,L)*COVAR(L,N2)
 1155 CONTINUE
 1160 CONTINUE
   10 FORMAT('1COEF = ',/(1X,10E12.5))
   11 FORMAT('1DELF = ',/(1X,10E12.5))
C
C
C     .   FILLING IN THOSE PARTS OF THE COEF MATRIX THAT DEPEND ON THE
C                 GYRO DRIFT RATES
C
      DO 1210 N = 1,6
      DO 1210 I = 1,2
 1210 COEF(I,N2-4+N) = XL(1,I,3+N)
      CALL DEQUIV(1,XL,DUMMY)
C
      DO 1200 I = 2,MSTOP
```

```
      CALL DEQUIV(I,PSI,DUM2)
      CALL DEQUIV(I,XL,DUM3)
      CALL DXMAT(DUM2,DUMMY,DUM4,3,6,6)
      CALL DADD(DUM4,DUM3,DUMMY)
      DO 1220 N = 1,3
      COEF(2 * I - 1,N2 - 4 + N) = DUMMY(1,3 + N)
 1220 COEF(2*I,N2-4+N) = DUMMY(2,3+N)
 1200 CONTINUE
C
      N2M = N2 - 1
C
C
C          DETERMINE VARIANCE FROM REPRESENTATION ERRORS
C
      DO 1250 I=1,N2
      VARCO(I) = 0.
      ACVAR(I) = 0.
      DO 1250 L=1,N2
      DO 1250 K=1,N2
      DIRTY = DEFL(I,L) * DEFL(I,K)
 1250 VARCO(I) = VARCO(I) + G**2 * DIRTY * COWT(L,K) * SEIN(L,K)
C
C          FIND VARIANCE DUE TO CORRELATED GYRO ERRORS.
C
      TIME(1) = 0.
      DO 1252 I=1,MSTOP
 1252 TIME(I+1) = TIME(I) + TSTOP(I) + TGO(I)
      TFIN = TIME(MSTOP+1)
      DO 1253 I=1,MSTOP
 1253 TIME(I) = 0.5 * (TIME(I) + TIME(I+1))
C
C          FIND VARIANCE DUE TO ACCELEROMETER ERRORS
C
      JTEST = 0
      DO 1261 I=1,2
      DO 1261 L=1,MSTOP
      DO 1261 K=1,MSTOP
 1261 COVAC(I,L,K) = ACSIG(I)* EXP(-ABS(TIME(L)-TIME(K))/ACTAV(I))
      DO 1256 I = 1,3
      TAUGY(I) = 0.5*TFIN
 1256 VARGY(I) = TARGY(I)*(1.-EXP(-TFIN/TAUGY(I)))
      WRITE(6,3)(VARGY(I),TAUGY(I),I=1,3)
 1254 CONTINUE
      DO 1255 I=1,3
      DO 1255 J=1,MSTOP
      DO 1255 K=1,MSTOP
 1255 COGY(I,J,K) = VARGY(I) * EXP(-ABS(TIME(J)-TIME(K))/TAUGY(I))
      IF(JTEST.NE.0)GO TO 1279
      DO 1270 I=1,3
      DO 1270 J=1,MSTOP
C
C
      CALL DEQUIV(J,XL,DUMMY)
```

```
      GYVAR(I,2*J-1,J) = DUMMY(1,I+3)
      GYVAR(I,2*J,J) = DUMMY(2,I+3)
C
      JPLUS = J + 1
C
      DO 1260 K = JPLUS,MSTOP
      CALL DEQUIV(K,PSI,DUM2)
      CALL DXMAT(DUM2,DUMMY,DUM3,6,6,6)
      CALL DEQUAL(DUM3,DUMMY)
C
      GYVAR(I,2*K-1,J) = DUMMY(1,I+3)
 1260 GYVAR(I,2*K,J) = DUMMY(2,I+3)
 1270 CONTINUE
      DO 1262 I=1,M2
      ACVAR(I) = 0.
      DO 1262 L=1,M2
      DO 1262 K=1,M2
      IF(MOD(L+K,2).NE.0) GO TO 1262
      L1 =(L+1)/2
      K1 =(K+1)/2
      MSS = 2 - MOD(L,2)
      ACVAR(I) = ACVAR(I) + COVAC(MSS,L1,K1) * DEFL(I,L) * DEFL(I,K)
 1262 CONTINUE
  116 FORMAT('1COVAC')
 1279 CONTINUE
      DO 1280 I=1,3
      DO 1280 J=1,M2
      GYRO(I,J) = 0.
C
      DO 1280 K=1,MSTOP
      DO 1280 L=1,MSTOP
 1280 GYRO(I,J) = GYRO(I,J) + GYVAR(I,J,K) * GYVAR(I,J,L) * COGY(I,K,L)
C
C         DEFINING THE MATRIX - SQMAT - WHICH SOLVED GIVES THE DESIRED
C                   LEAST SQUARES SOLUTION FOR THE DEFLECTIONS OF THE
C                   VERTICAL AND THE GYRO DRIFT RATES.
C                   THE BULK OF SQMAT IS THEN THE ERROR COVARIANCE MATRIX
C
      DO 1282 I=1,M2
      WATE(I) = ACVAR(I) + VARCO(I)
      DO 1283 K=1,3
 1283 WATE(I) = WATE(I) + GYRO(K,I)
 1282 CONTINUE
      TOTSIG = 0.
      SUMWT = 0.
      DO 1284 J=1,M2
      TOTSIG = TOTSIG + WATE(J)
 1284 SUMWT = SUMWT + 1./WATE(J)
      SIGWT = SUMWT/FLOAT(M2)
      WRITE(6,122)TOTSIG,M2,SUMWT
  122 FORMAT(1H1,'VARIANCES OF INDIVIDUAL POINTS',//'0TOTAL VARIANCES ='
     *    ,E15.7,/' NUMBER OF POINTS =',I4,/' SUMWT = ',E15.7)
      WRITE(6,123)
```

```
  123 FORMAT(////T3,'STEP',T14,'AC.VARIANCE',T37,'COL.VARIANCE',
     *T55,'VERTICAL',T79,'NORTH',T99,'EAST' ,T113,'TOTAL VARIANCE')
      DO 1286J=1,M2
 1286 WRITE(6,125)J,ACVAR(J),VARCO(J),(GYRO(I,J),I=1,3),WATE(J)
  125 FORMAT(I8,6E20.8)
  119 FORMAT(    '0VARIANCES')
  120 FORMAT(     1X,10E12.4)
  121 FORMAT('0GYRO JUNK')
      DO 1285 J=1,M2
 1285 IF(IWATE.EQ.0) WATE(J) = 1.0
      IF(IWATE.EQ.0)SUMWT = FLOAT(M2)
      DO 1292 I=1,N2
      COMEAN(I) = 0.
      DO 1289 J=1,M2
 1289 COMEAN(I) = COMEAN(I) + COEF(J,I) / WATE(J)
 1292 COMEAN(I) = COMEAN(I)/SUMWT
      N2M2 = N2M
      N22 = N2
      IF(JTEST.EQ.2) N22 = N2 - 3
      IF(JTEST.EQ.2) N2M2 = N2M-3
      DO 1300 I = 1,N2M2
      DO 1300 J = 1,N22
      SQMAT(I,J) = 0.
      DO 1300 N = 1,M2
 1300 SQMAT(I,J) = SQMAT(I,J) + (COEF(N,I)-COMEAN(I))*(COEF(N,J)
     +             - COMEAN(J))  / WATE(N)
      IF(JTEST.EQ.0)GO TO 1302
      IF(JTEST.EQ.2)GO TO 1303
      N2M2 = N2M-1
      N22 = N2 - 1
      DO 1304 J=1,2
      DO 1304 I=1,NN
 1304 SQMAT(I,J+NN) = SQMAT(I,J+NN+1)
      DO 1305 I=1,2
      DO 1305 J=1,NN
 1305 SQMAT(I+NN,J) = SQMAT(J,I+NN)
      DO 1306 I=1,2
      DO 1306 J=1,2
 1306 SQMAT(I+NN,J+NN) = SQMAT(I+NN+1,J+NN+1)
 1303 CONTINUE
      DO 1301 I=1,N2M
      SQMAT(I,N22)=0.
      DO 1301 J=1,M2
 1301 SQMAT(I,N22) = SQMAT(I,N22)+ (COEF(J,I)-COMEAN(I))*(COEF(J,N2)
     *    -COMEAN(N2))/WATE(J)
      IF(JTEST.EQ.2)GO TO 1302
      SQMAT(NN+1,N22) = SQMAT(NN+2,N22)
      SQMAT(NN+2,N22) = SQMAT(NN+3,N22)
 1302 CONTINUE
C
C         AFTER SUBROUTINE SOLVE, SQMAT(*,2*N) CONTAINS THE SOLUTION
C                   VECTOR, THE LAST  THREE ARE THE GYRO RATES AND THE
C                   REST ARE THE DEFLECTIONS OF THE VERTICAL.
```

```
C                 THE LAST COLUMN IS EQUIVALENCED TO A VECTOR SOMAT
C
      WRITE(6,12) SOMAT
   12 FORMAT(*1SOMAT = */(1X,5E12.5))
      CALL SOLVE (SOMAT,N2,N2M2,SOMAT)
      DO 1350 J=1,N22
 1350 WRITE(6,12)  (SOMAT(I,J),I=1,N212)
  140 FORMAT(*1SOMAT = *,/(6F18.7))
      WRITE(6,12)SOMAT
      DO 1400 I = 1,M2
      F(I) = 0.
C
C          DETERMINATION OF THE ACTUAL VARIANCE OF THE SOLUTION
      DO 1400 J = 1,N2M2
 1400 F(I) = F(I) + COEF(I,J) * SOMAT(J)
      SUMSQ=0.
      DO 1450 I = 1,M2
      VAR(I) = (F(I) - COEF(I,N2))**2
 1450 SUMSQ = SUMSQ + VAR(I)
      SIGMA = SQRT(SUMSQ/FLOAT(M2-1))
      NN = N2 - 4
      DO 1460 I = 1,NN
 1460 SOMAT(I) = SOMAT(I)/G
C
C
C          OUTPUT THE FINAL RESULTS
C
      IF(JTEST.EQ.0)
     *WRITE(6,110) SOMAT(NN+1),SOMAT(NN+2),SOMAT(NN+3)
  110 FORMAT(1H1,* FINAL RESULTS *///*GYRO DRIFT RATES,    ALPHA*,E12.4,5X
     2,*BETA*,E12.4,5X,*GAMMA*,E12.4//*DEFLECTIONS OF VERT*/* XI*
     3 ,17X,*ETA*,17X*NORTH POS*,11X,*EAST POS*//)
      IF(JTEST.EQ.1)WRITE(6,131)SOMAT(NN+1),SOMAT(NN+2)
  131 FORMAT(1H1,* RESULTS WITHOUT VERTICAL GYRO DRIFT*//
     +    * GYRO DRIFT RATES,   BETA*,E12.4,5X,*GAMMA*,E12.4//
     +    * DEFLECTIONS OF VERT*/* XI*,17X,*ETA*,17X,*NORTH POS*,11X,
     +    *EAST POS*//)
      IF(JTEST.EQ.2)WRITE(6,130)
      N1M = N1 - 1
 1600 WRITE (6,111) XIO,ETAO,XHAT(1),YHAT(1)
      WRITE (6,111) (SOMAT(2*I-3),SOMAT(2*I-2),XHAT(I),YHAT(I),I=2,N1M)
      WRITE(6,111) XIFIN,ETAFIN,XHAT(N1),YHAT(N1)
  111 FORMAT (1X,4(E16.8,4X))
      WRITE(6,112)SUMSQ,SIGMA,(I,VAR(I),I = 1,M2)
  112 FORMAT(1H1,*VARIANCE OF SOLUTION*,E12.4,5X,*SIGMA*,E12.4/
     * *INDIVIDUAL LSQ ERRORS*/(1X,I5,5X,E12.4))
      IF(JTEST.EQ.2)GO TO 1650
      IF(JTEST.EQ.1)GO TO 1501
      JTEST = 1
      VARGY(1) = TARGY(1)
      TAUGY(1) = TAUY(1)
      GO TO 1254
 1501 JTEST = 2
```

```
      DO 1502 I=2,7
      VARGY(I) = TARGY(I)
 1502 TAUGY(I) = TAUTY(I)
  10  FORMAT(1H1,' RESULTS WITHOUT GYRO DRIFTS'//
     *    '  DEFLECTIONS OF VERT'/' XI',17X,'ETA',17X,'NORTH POS',11X,
     *    'EAST POS'//)
      IF(JTEST.NE.0) GO TO 1250
 1650 CONTINUE
      STOP
      END
```

```
      SUBROUTINE COLLOC(M,N1,COVAR,X,Y,XHAT,YHAT,COWT)
C
C        THIS SUBROUTINE  PRODUCES A MATRIX - COVAR - THAT PRODUCES
C                 VALUES FOR THE DEFLECTION OF THE VERTICAL AT POINTS, I,
C                 FROM THE VALUES OF THE DEFLECTIONS AT OTHER POINTS, J.
C        THIS IS DONE BY STATISTICAL COLLOCATION.
C        FOR A DERIVATION OF THE METHOD SEE THE PHOENIX CORP. REPORT
C
      COMMON/SCRTCH/CV1,CVINV,CV2,FILL(48)
      DIMENSION CVTEST(24,24)
      DIMENSION CV1(24,24),CVINV(24,24),CV2(80,80),COVAR(80,24),L2(24),
     2 X(50),Y(50),XHAT(50),YHAT(50),MM(24)
      DIMENSION CONT(80,80)
      DIMENSION CDUM(576)
      EQUIVALENCE(CDUM(1),CVINV(1,1) )
      SIGD2 = SIGD(1.,1.) **2
      SIGG2 = SIGMA(1.,1.)**2
C
C        DETERMINE THE COVARIANCES BETWEEN THE DEFLECTIONS AT THE
C                 BASIC POINTS.  ODD INICES  DENOTE XI VALUES, EVEN
C                 INDICES  ETA VALUES. . THE COVARIANCES ARE DERIVED
C                 UNDER THE ASSUMPTION OF ISOTROPIC, HOMOGENEOUS
C                 COVARIANCE OF THE GRAVITY ANOMALY.
C
      DO 500 I = 1,N1
      DO 500 J = 1,N1
      R= SQRT((XHAT(I)-XHAT(J))**2 + (YHAT(I) - YHAT(J))**2)
      IF (R.EQ.0.)GO TO 498
      WRITE(6,499)R,YHAT(I),XHAT(J),YHAT(I),YHAT(J)
  499 FORMAT('0',5E12.5)
      STH = (YHAT(I) - YHAT(J))/R
      CTH = (XHAT(I) - XHAT(J))/R
  498 IF(R.EQ.0.)STH=0.
      IF(R.EQ.0.)CTH=0.
      CV1(2*I-1,2*J-1)=SIGD2*(PHIGG(1.,R)/SIGG2+(STH**2-CTH**2)*FC(1.,R)
     2)
      CV1(2*I,2*J) = SIGD2*(PHIGG(1.,R)/SIGG2+(CTH**2- STH**2)*FC(1.,R))
      CV1(2*I-1,2*J)=        -2. * SIGD2 * STH * CTH * FC(1.,R)
      CV1(2*I,2*J-1) = CV1(2*I-1,2*J)
  500 CONTINUE
  490 FORMAT('1','CV1',/,(8E14.5))
C
C
C                 INVERSION OF THE CV1 MATRIX  IS FOUND IN CVINV
C
      M2 = 2 * M
      ILNTH = 4 * N1 * N1
      N12 = 2 * N1
      DO 491 J=1,N12
      DO 491 I=1,N12
  491 CDUM(I+(J-1)*N12) = CV1(I,J)
      CALL MINV(CVINV,N12,D,L2,MM,ILNTH)
  492 FORMAT('1INVERSE OF CV1',/,(8E14.5))
```

```
      DO 493 J=1,N12
      DO 493 I=1,N12
  493 CV1(I,J) = COUM(I+(J-1) * N12)
      DO 494 I=1,N12
      DO 494 J=1,N12
  494 CVINV(I,J) = CV2(I,J)
  490 FORMAT  (*1CV1 AND INVERSE TEST PRODUCT*,/(1X,8E16.8))
      DO 600 K = 1,M
C
C         DETERMINATION OF THE COVARIANCES BETWEEN THE BASIS SET AND
C                 THE SET OF POINTS DETERMINED BY THE MISSION LEGS.
C
      DO 600 I = 1,M1
      R = SQRT((X(K) - XHAT(I))**2 + (Y(K)-YHAT(I))**2)
      IF(R.EQ.0.)GO TO 550
      STH =(Y(K) - YHAT(I))/R
      CTH = (X(K) - XHAT(I))/R
  550 IF(R.EQ.0.)CTH=0.
      IF (R.EQ.0.)STH=0.
      CV2(2*K-1,2*I-1) = SIGG2 *(PHIGG(1.,R)/SIGG2 +(STH**2- CTH**2)
     *    * FC(1.,R))
      CV2(2*K,2*I) = SIGG2*(PHIGG(1.,R)/SIGG2 + (CTH**2 - STH**2)
     *    * FC(1.,R))
      CV2(2*K-1,2*I) = -2.*SIGG2*STH*CTH*FC(1.,R)
      CV2(2*K,2*I-1) = CV2(2*K-1,2*I)
  600 CONTINUE
      N2 = 2*M1
      M2 = 2*M
C
C         PRODUCTION OF THE COVAR MATRIX BY MULTIPLICATION OF CV2 BY
C                 CVINV
C
      DO 700 L = 1,M2
      DO 700 K = 1,N2
      COVAR(L,K) = 0.
      DO 650 I = 1,N2
  650 COVAR(L,K) = COVAR(L,K) + CV2(L,I)*CVINV(I,K)
  700 IF(MOD(L,2).NE.0) COVAR(L,K) = -COVAR(L,K)
      DO 810 K=1,M
      DO 810 I=1,M
      R = SQRT((X(K) - X(I))**2 + (Y(K) - Y(I))**2)
      IF(R.EQ.0.) GO TO 820
      STH = (Y(K) - Y(I))/R
      CTH = (X(K) - X(I))/R
      GO TO 821
  820 CTH = 0.
      STH = 0.
  821 COWT(2*K-1,2*I-1) = SIGG2 * (PHIGG(1.,R)/SIGG2
     *    + (STH**2 - CTH**2) * FC(1,R))
      COWT (2*K,2*I) = SIGG2 * (PHIGG(1.,R)/SIGG2
     *    + (CTH**2 - STH**2) * FC(1.,R))
      COWT(2*K-1,2*I) = -2.*SIGG2 * STH * CTH * FC(1.,R)
      COWT(2*K,2*I-1) = COWT(2*K-1,2*I)
```

```
  810 CONTINUE
      DO 800 L1 = 1,M2
      DO 800 L2 = 1,M2
      DO 800 K = 1,M2
      SIGN = 1.
      IF(MOD(L2,2).NE.0) SIGN = -1.
  800 CONT(L1,L2) = CONT(L1,L2) - CV2(L1,K) * COVAR(L2,K) * SIGN
      RETURN
   10 FORMAT(*1COVAR = *,/(1X,10E12.5))
      END
```

```
      FUNCTION SIGMA(X,R)
C
C        THIS FUNCTION AND ITS ENTRIES GIVE THE NECESSARY VALUES FOR
C              THE COMPUTATION  OF THE DEFLECTION COVARIANCES.
C        CURRENTLY, THE FUNCTION ASSUMES A SECOND ORDER MARKOVIAN
C              STRUCTURE FOR THE ANOMALY COVARIANCE.  VAR IS THE
C              VARIANCE OF THE ANOMALY  AND D IS THE CORRELATION
C              LENGTH
C
      DATA VAR,D,G,ROOT2/7.30E-2,4.E6,9.80665E2,1.41421/
      SIGMA = VAR
      RETURN
C     ENTRY SIGD
      ENTRY SIGD
C     ENTRY SIGD(X,R)
      SIGMA= VAR/G/ROOT2
      RETURN
C     ENTRY PHIGG (X,R)
C     ENTRY PHIGG
      ENTRY PHIGG
      Q = R/D
      SIGMA =VAR**2* EXP(-Q)*(1.+Q)
      RETURN
C     ENTRY FC (X,R)
C     ENTRY FC
      ENTRY FC
      IF(R.EQ.0.) GO TO 10
      Q = R/D
      SIGMA = 6./Q**2 - EXP(-Q)* (Q+3. + 6./Q**2 +6./Q)
      RETURN
   10 SIGMA = 0.
      RETURN
      END
```

```
      SUBROUTINE ADVANS(T1,T2,PSI,XL,N,XPOS,YPOS)
C        THIS SUBROUTINE PROVIDES THE VALUES OF THE NECESSARY TIME
C                  SHIFT MATRICES.
C        THIS ENTRY INITIALIZES THE VALUES OF THE NECESSARY
C                  EIGENVECTOR MATRICES
C
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION LAM2,LAM2I,PRODR(6,6),PRODI(6,6)
      REAL T1,T2,         XPOS,YPOS,CPHI1,SPHI1,TPHI1
      REAL SNGL
      REAL REAL
      DIMENSION XPOS(50),YPOS(50)
C     COMPLEX A
      COMPLEX B
      COMPLEX CTEMP
      COMPLEX CEXP
C     COMPLEX LAM2
      COMPLEX TESTY(6,6)
      COMPLEX TESTY(3,3)
      COMMON CPHI1,SPHI1,TPHI1
      COMPLEX R(6),P(2),C(6),C2(6),D(2),P2(2),S(6,6),SINV(6,6),T(3,3),
     2 TINV(3,3),PH(6,6),PHI(6,6),VL(6,6),VLI(6,6),PO(3,3),VP(3,3)
      DIMENSION C2R(6),C2I(6),PHR(6,6),PHM(6,6),PHIR(6,6),PHII(6,6),
     1          VLR(6,6),VLM(6,6),VLIR(6,6),VLII(6,6)
      DIMENSION SI(6,6),SR(6,6) ,SINVR(6,6),SINVI(6,6)
      DIMENSION RR(6),RI(6),CR(6),CI(6)
      DIMENSION RPHI(6,6),RVLI(6,6),RPO(3,3),RVP(3,3),RPHO(6,6),
     1RVPO(6,6),
     2 DUMMY(6,6),DUMB(6,6),PSI(50,6,6),XL(50,6,6)
      DATA G,REARTH,OMEGA/9.80665D2,6.37103D8,7.29212D-5/
      NAMELIST/DBG1/P,RI,DI,OMEGAS ,RR,RI
      NAMELIST/DBUG/B,SR,DI,CTEMPR,CTEMPI,I,K        ,CTEMP
      NAMELIST/DBG2/S,SINV,A,B,G,REARTH
      NAMELIST/DBG3/TINV    ,P
      OMEGAS =DSQRT(G/REARTH)
      CPHI = CPHI1
      SPHI = DSQRT(1. - CPHI*CPHI)
      TPHI = SPHI/CPHI
      DO 200 I = 1,6
      DO 200 J = 1,6
         RVPO(I,J) = 0
  200 RPHO(I,J) = 0.
      DO 210 I = 1,6
  210 RPHO(I,I) = 1.
C
C        FIRST FOR THE TIME WHEN THE VEHICLE IS IN MOTION
C
C
C        P CONTAINS THE ROOTS OF THE SECULAR EQUATION
C
      OMEGA2 = OMEGA * OMEGA
```

```
      CPHI2 = CPHI * CPHI
      SPHI2 = SPHI * SPHI
      R(1) = (0.,1.) * OMEGAS
      R(2) =-R(1)
      A1 = OMEGA**2 + OMEGAS**2
C     B1 = SQRT(OMEGA**4 +OMEGAS**4+2.*OMEGA**2*OMEGAS**2*(1.+2.*CPHI2))
      B1 =DSQRT(OMEGA**4 +OMEGAS**4+2.*OMEGA**2*OMEGAS**2*(1.+2.*CPHI2))
C     R(3) = (0.,1.) * SQRT(0.5 * A1 + 0.5 * B1)
      R(3) = (0.,1.) *DSQRT(0.5 * A1 + 0.5 * B1)
      R(4) = - R(3)
      R(5) = (1.,0.)*DSQRT(0.5*B1-0.5*A1)
      R(6) = -R(5)
C
      RR(1) = 0.
      RI(1) = OMEGAS
C
      RR(2) = 0.
      RI(2) = OMEGAS * (-1.)
C
      RR(3) = 0.
C     RI(3) = SQRT(.5 * A1 + 0.5 * B1)
      RI(3) =DSQRT(.5 * A1 + 0.5 * B1)
C
       RR(4)= 0.
      RI(4) = -RI(3)
      RR(5) =DSQRT(0.5 * B1 -.5 * A1)
C     RR(5) = SQRT(0.5 * B1 -.5 * A1)
      RI(5) = 0.
C
      RR(6) = -RR(5)
      RI(6) = 0.
C
C        FILL IN THE VALUES FOR THE TRANSFORMATION MATRIX, S, AND
C                  ITS INVERSE, SINV
C
      DO 500 I = 1,6
      LAM2 = R(I) * R(I)
      LAM2I = 0.
      A = (R(I)**2 + OMEGAS**2)
      AI = 0.
      B =     REARTH * OMEGA * SPHI * (-3. * LAM2**3 - LAM2*LAM2 *
     1   (2. * OMEGA2 * OMEGAS + OMEGA2)+ LAM2 * (OMEGAS**4 + OMEGAS**2
     2   * OMEGA2 * SPHI2) - 3. * OMEGAS**4 * OMEGA2 * CPHI2)
      CALL TIMES(RR(I),RI(I),RR(I),RI(I),LAM2,LAM2I)
      A = OMEGAS * OMEGAS + LAM2
      AI = 0.
      S(I,1) = R(I) * OMEGA * SPHI * (-OMEGA2 * CPHI2 + LAM2)
      S(I,2) = LAM2 * (OMEGA2 + OMEGAS * OMEGAS + LAM2)
     1   - OMEGAS * OMEGAS * OMEGA2 * CPHI2
      S(I,3) = -LAM2 * OMEGA2 * OMEGA * SPHI * CPHI2
      S(I,4) = REARTH * LAM2 * R(I) * OMEGA2 * SPHI * CPHI
      S(I,5) = REARTH * R(I) * ((LAM2 - OMEGAS * OMEGAS) * OMEGA2 *
     1   SPHI2 + LAM2 * (OMEGAS * OMEGAS + LAM2))
```

```
      SI(I,J) = - REARTH * LAM2 * LAM2 * OMEGA * SPHI
      SINV(1,I) = G*A/R(I)
      SINV(2,I) = G*OMEGA*SPHI
      SINV(3,I) = G*A/R(I)**2
      SINV(4,I) = -(A**2 + (R(I)*OMEGA*SPHI)**2)/(R(I)*OMEGA*CPHI)
      SINV(5,I) = -R(I) * OMEGA*SPHI
      SINV(6,I) = A
C
      CONST = OMEGA * SPHI * (-OMEGA2 * CPHI2 + LAM2)
C
      SR(I,1) = RR(I) * CONST
      SI(I,1) = RI(I) * CONST
C
      SR(I,2) = LAM2 * (OMEGA2 + OMEGAS * OMEGAS + LAM2)
     1   - OMEGAS * OMEGAS * OMEGA2 * CPHI2
      SI(I,2) = 0.
      SR(I,3) = -LAM2 * OMEGA2 * OMEGA * SPHI * CPHI2
      SI(I,3) = 0.
C
      SR(I,4)= REARTH * LAM2 *RR(I) * OMEGA2 * SPHI * CPHI
      SI(I,4)= REARTH * LAM2 *RI(I) * OMEGA2 * SPHI * CPHI
      SR(I,5)= REARTH *RR(I) * ((LAM2 - OMEGAS * OMEGAS) * OMEGA2 *
     1   CPHI2 + LAM2 * (OMEGAS * OMEGAS + LAM2))
      SI(I,5)= REARTH *RI(I) * ((LAM2 - OMEGAS * OMEGAS) * OMEGA2 *
     1   CPHI2 + LAM2 * (OMEGAS * OMEGAS + LAM2))
      SR(I,6)= - REARTH * LAM2 * LAM2 * OMEGA * SPHI
      SI(I,6) = 0.
      SINVR(1,I) = G*A
      CALL DIVIDE(SINVR(1,I),0.0,RR(I),RI(I),SINVR(1,I),SINVI(1,I))
C
      SINVR(2,I) = G * OMEGA * SPHI
      SINVI(2,I) = 0.
C
      SINVR(3,I) = G * A / LAM2
      SINVI(3,I) = 0.
C
      SINVR(4,I) = OMEGA * SPHI * RR(I)
      SINVI(4,I) = OMEGA * SPHI * RI(I)
      CALL TIMES(SINVR(4,I),SINVI(4,I),SINVR(4,I),SINVI(4,I),
     1         SINVR(4,I),SINVI(4,I))
      SINVR(4,I) =-(SINVR(4,I) + A*A )
      DTEMPR = RR(I) * OMEGA * CPHI
      DTEMPI = RI(I) * OMEGA * CPHI
      CALL DIVIDE(SINVR(4,I),SINVI(4,I),DTEMPR,DTEMPI,SINVR(4,I),
     1         SINVI(4,I))
C
      SINVR(5,I) = - RR(I) * OMEGA * SPHI
      SINVI(5,I) =-RI(I) * OMEGA * SPHI
C
      SINVR(6,I) = A
      SINVI(6,I) = 0.
C
      BR = 0.
```

```
      BT = 0.
      B = (0.,0.)
      AMAZE = 0.
      DO 497 K=1,6
      AX = DSQRT(SR(I,K)**2 + SI(I,K)**2)
  497 AMAZE = DMAX1(AX,AMAZE)
      DO 493 K=1,6
      CALL DIVIDE(SR(I,K),SI(I,K),AMAZE,0.0D0,SR(I,K),SI(I,K))
  493 S(I,K) = S(I,K)/AMAZE
      DO 499 K=1,6
      CALL TIMES(SINVR(K,I),SINVI(K,I),SR(I,K),SI(I,K),DTEMPR,DTEMPI)
      BR = BR + DTEMPR
      BI = BI + DTEMPI
      DTEMP = SINV(K,I) * S(I,K)
      B = B + DTEMP
  499 CONTINUE
      SINV(1,I) = SINV(1,I)/B
      SINV(2,I) = SINV(2,I)/B
      SINV(3,I) = SINV(3,I)/B
      SINV(4,I) = SINV(4,I)/B
      SINV(5,I) = SINV(5,I)/B
      SINV(6,I) = SINV(6,I)/B
      CALL DIVIDE(SINVR(1,I),SINVI(1,I),BR,BI,SINVR(1,I),SINVI(1,I))
      CALL DIVIDE(SINVR(2,I),SINVI(2,I),BR,BI,SINVR(2,I),SINVI(2,I))
      CALL DIVIDE(SINVR(3,I),SINVI(3,I),BR,BI,SINVR(3,I),SINVI(3,I))
      CALL DIVIDE(SINVR(4,I),SINVI(4,I),BR,BI,SINVR(4,I),SINVI(4,I))
      CALL DIVIDE(SINVR(5,I),SINVI(5,I),BR,BI,SINVR(5,I),SINVI(5,I))
      CALL DIVIDE(SINVR(6,I),SINVI(6,I),BR,BI,SINVR(6,I),SINVI(6,I))
  500 CONTINUE
  455 FORMAT(//1H )
  495 FORMAT(1X,2I4,10X,4E14.7)
      CALL CXMAT (S,SINV,TESIX,6)
      CALL CXMAT(SINV,S,TESIX,6)
      CALL COMPRD(SR,SI,SINVR,SINVI,PRODR,PRODI,6,6,6)
      CALL COMPRD(SINVR,SINVI,SR,SI,PRODR,PRODI,6,6,6)
C
C        NOW DO THE SAME FOR WHEN THE VEHICLE IS STOPPED
C
      P(1) = (0.,1.)* OMEGA
      P(2) = -P(1)
      A1 = 1./DSQRT(2.D0)
      T(1,1) = -CPHI * A1
      T(1,2) = SPHI * A1
      T(1,3) = (0.,1.)*A1
      T(2,1) = T(1,1)
      T(2,2) = T(1,2)
      T(2,3) = - T(1,3)
      T(3,1) = SPHI
      T(3,2) = CPHI
      T(3,3) = 0.
      TINV(1,1) = -CPHI * A1
      TINV(1,2) = T(1,1)
      TINV(1,3) = SPHI
```

```
      TINV(2,1) = SPHI * A1
      TINV(2,2) = TINV(2,1)
      TINV(2,3) = CPHI
      TINV(3,1) = -(0.,1.) * A1
      TINV(3,2) = -TINV(3,1)
      TINV(3,3) = 0
  601 FORMAT(*1T = *,(1X,6E12.4))
  602 FORMAT(*1TINV = *,/(1X,6E12.4))
      CALL CXMAT(T,TINV,TESTY,3)
      CALL CXMAT(T,TINV,TESTY,3)
      CALL CXMAT(TINV,T,TESTY,3)
  607 FORMAT(*1T AND TINV PRODUCT = *,/(1X,6E12.4))
      RETURN
C     ENTRY TIMEX
      ENTRY TIMEY
C     ENTRY TIMEX(T1,T2,PSI,XL,N,XPOS,YPOS)
C
C        THIS ENTRY ACTUALLY CALCULATES THE TIME SHIFT MATRICES
C                T1 IS THE TIME THE VEHICLE IS MOVING
C                T2 IS THE TIME STOPPED
C                THE DERIVATION OF THE VARIOUS OPERATIONS PERFORMED
C                HERE IS FOUND IN THE PHOENIX CORP. REPORT
C
      DT = T1 * 0.5
      DO 600 K = 1,5
C
      DTEMPR = PR(K) * DT
      DTEMPI = PI(K) * DT
C
      CR(K) = DEXP(DTEMPR) * DCOS(DTEMPI)
      CI(K) = DEXP(DTEMPR) * DSIN(DTEMPI)
C
      CALL TIMTS(CR(K),CI(K),CR(K),CI(K),C2R(K),C2I(K))
C
      C(K) = CEXP(P(K) * SNGL(DT))
      C2(K) = C(K) * C(K)
C
  600 CONTINUE
      DO 700 I = 1,5
      DO 700 J = 1,5
      PHR(I,J) = 0.
      PHR(I,J) = 0.
      PHIR(I,J) = 0.
      PHII(I,J) = 0.
      VLR(I,J) = 0.
      VLR(I,J) = 0.
      VLIR(I,J) = 0.
      VLII(I,J) = 0.
      PH(I,J) = (0.,0.)
      PHI(I,J) =(0.,0.)
      VL(I,J) = (0.,0.)
      VLI(I,J) = (0.,0.)
      DO 700 K = 1,6
```

```
C
C
      CALL TIMES(C2R(K),C2I(K),SINVR(I,K),SINVI(I,K),DTEMPR,DTEMPI)
      CALL TIMES(DTEMPR,DTEMPI,SR(K,J),SI(K,J),DTEMPR,DTEMPI)
      PHIR(I,J) = PHIR(I,J) + DTEMPR
      PHII(I,J) = PHII(I,J) + DTEMPI
C
      PHI(I,J) = PHI(I,J) + C2(K)*SINV(I,K)*S(K,J)
C
C
      CALL TIMES(RR(K),RI(K),CR(K),CI(K),DTEMPR,DTEMPI)
      CALL TIMES(DTEMPR,DTEMPI,SINVR(I,K),SINVI(I,K),DTEMPR,DTEMPI)
      CALL TIMES(DTEMPR,DTEMPI,SR(K,J),SI(K,J),DTEMPR,DTEMPI)
      PHR(I,J) = PHR(I,J) + DTEMPR
      PHM(I,J) = PHM(I,J) + DTEMPI
C
      PH(I,J) = PH(I,J) + R(K) * C(K)*SINV(I,K)*S(K,J)
  830 FORMAT(*0PH      *,4E21.7)
C
      DTEMPR = C2R(K) - 1.
      CALL DIVIDE(DTEMPR,C2I(K),RR(K),RI(K),DTEMPR,DTEMPI)
      CALL TIMES(DTEMPR,DTEMPI,SINVR(I,K),SINVI(I,K),DTEMPR,DTEMPI)
      CALL TIMES (DTEMPR,DTEMPI,SR(K,J),SI(K,J),DTEMPR,DTEMPI)
      VLIR(I,J) = VLIR(I,J) + DTEMPR
      VLII(I,J) = VLII(I,J) + DTEMPI
C
      VLI(I,J) = VLI(I,J) + (C2(K)-(1.,0.))/R(K)*SINV(I,K)*S(K,J)
C
C
      CALL TIMES (CR(K),CI(K),SINVR(I,K),SINVI(I,K),DTEMPR,DTEMPI)
      CALL TIMES(DTEMPR,DTEMPI,SR(K,J),SI(K,J),DTEMPR,DTEMPI)
      VLR(I,J) = VLR(I,J) + DTEMPR
      VLM(I,J) = VLM(I,J) + DTEMPI
C
      VL(I,J) = VL(I,J) +(C(K)*SINV(I,K)*S(K,J))
C
  820 FORMAT(*0VL(*,I1,****,I1,*(*,4E21.7)
C
C
  700 CONTINUE
  415 FORMAT(*0*,6E21.7)
C
C
      DO 800 I = 1,5
C
      PHIR(1,I) = PHIR(1,I) - PHR(4,I) * (YPOS(N+1) -  YPOS(N))
      PHII(1,I) = PHII(1,I) - PHM(4,I) * (YPOS(N+1) - YPOS(N))
      PHIR(2,I) = PHIR(2,I) + PHR(4,I) * (XPOS(N+1) - XPOS(N))
      PHII(2,I) = PHII(2,I) + PHI(4,I) * (XPOS(N+1) - XPOS(N))
C
      VLIR(1,I) = VLIR(1,I) - VLR(4,I) * (YPOS(N+1) - YPOS(N))
      VLII(1,I) = VLII(1,I) - VLM(4,I) * (YPOS(N+1) - YPOS(N))
      VLIR(2,I) = VLIR(2,I) + VLR(4,I) * (XPOS(N+1) - XPOS(N))
```

```
      VLI(2,I) = VLI(2,I) + VLI(4,I) *(XPOS(N+1) - XPOS(N))
C
      PHI(1,I) = PHI(1,I) - PH(4,I)*(YPOS(N+1)-YPOS(N))
      PHI(2,I) = PHI(2,I) + PH(4,I)*(XPOS(N+1)-XPOS(N))
      VLI(1,I) = VLI(1,I) - VL(4,I) * (YPOS(N+1) - YPOS(N))
  800 VLI(2,I) = VLI(2,I) + VL(4,I) * (XPOS(N+1) - XPOS(N))
  805 FORMAT(*1VLI =*/(1X,12E11.4))
  804 FORMAT(*1VL = */(1X,12E11.4))
  806 FORMAT (*1PH  = */(1X,12E11.4))
  807 FORMAT(*1PHI = */(1X,12E11.4))
      DO 810 I = 1,6
      DO 810 J=1,6
C     RPHI(I,J) = REAL(PHI(I,J))
      RPHI(I,J) = PHIR(I,J)
C 810 RVLI(I,J) = REAL(VLI(I,J))
  810 RVLI(I,J) = VLIR(I,J)
      D(1) = CEXP(P(1) * T2)
      D(2)=CEXP(P(2) * T2)
      DO 900 I = 1,3
      DO 900 J = 1,3
      PD(I,J) = D(1) * TINV(I,1)*T(1,J) + D(2) * TINV(I,2)*T(2,J)
     2    + TINV(I,3) * T(3,J)
  900 VP(I,J) = (D(1)-1.)/P(1)*TINV(I,1)*T(1,J)
     2 +(D(2)-1.) / P(2) * TINV(I,2) * T(2,J) + T2 * TINV(I,3) * T(3,J)
      DO 910 I = 1,3
      DO 910 J = 1,3
      RPD(I,J) = REAL(PD(I,J))
  910 RVP(I,J) = REAL(VP(I,J))
      DO 920 I = 1,3
      RPHD(3+I,3) = - OMEGA*UPHI/REARTH*RVP(I,1)
      DO 920 J = 1,3
      RVPD(3+I,3+J) = RVP(I,J)
  920 RPHD(3+I,3+J) = RPD(I,J)
      CALL DXMAT(RPHI,RPHD,DUMMY,6,6,6)
      CALL DEQUI(N,PSI,DUMMY)
      DO 950 I = 1,3
      DO 950 J = 1,3
  950 PSI(N,I,J) = 0.
      CALL DXMAT(RPHI,RVPD,DUMMY,6,6,6)
      CALL DADD(RVLI,DUMMY,DUM2)
      CALL DEQUI(N,VL,DUM2)
      RETURN
      END
```

SYMBOLIC REFERENCE MAP (R=3)

	DEF LINE	REFERENCES
	1	427
EA	227	367

```
      SUBROUTINE SOLVE (SQMAT,N2,N2M,SOMAT)
      DOUBLE PRECISION D,SOMAT,DUMMY,DUM2,SQMAT
      DIMENSION SQMAT(23,24 ),SOMAT(24)
      DIMENSION COL2(23),L(23),M(23),COLUMN(23),DUMMY(529)  ,DUM2(23,23)
      DO 10 I=1,N2M
      DO 10 J=1,N2M
   10 DUMMY(I + (J-1) * N2M) = SQMAT(I,J)
      DO 25 I=1,N2M
   25 COLUMN(I) = SQMAT(I,N2)
      ILNTH = N2M * N2M
      CALL DMINV(DUMMY,N2M,D,L, M,ILNTH)
      DO 200 I=1,N2M
      COL2( I ) = 0.
      DO 200 J=1,N2M
  200 COL2(I) = COL2(I) + DUMMY(I+(J-1) * N2M)* COLUMN(J)
      DO 300 I = 1,N2M
      SOMAT(I) = COL2(I)
  300 SQMAT(I,N2)=COL2(I)
      DO 350 I=1,N2M
      DO 350 J=1,N2M
  350 SQMAT(I,J) = DUMMY(I + (J-1) *N2M)
      IX = 23
      IY = 24
      CALL DXMAT(SQMAT,DUMMY,DUM2,N2M,IX,IY)
      WRITE(6,20) DUMMY
   20 FORMAT(*1SQMAT = *,/(1X,5E15.8))
      WRITE(6,30)DUM2
   30 FORMAT(*1PRODUCT OF SQMAT AND INVERSE = *,/(1X,5E16.8))
      RETURN
      END
```

```
C
C        MANY MATRIX OPERATIONS NOT FOUND IN FORTRAN
C
C        EQUIV  SETS A SMALL 6X6 MATRIX EQUAL TO A SECTION A LARGER
C                 ONE, EQUI IS THE INVERSE
C        ADD ADDS TWO MATRICES, XMAT MULTIPLIES TWO MATRICES AND
C        EQUAL SETS ONE MATRIX EQUAL TO ANOTHER
C
      SUBROUTINE EQUIV(N,A,B)
      DIMENSION A(50,6,6),B(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 B(I,J) = A(N,I,J)
      RETURN
      END


      SUBROUTINE DEQUIV(N,A,B)
      DOUBLE PRECISION A,B
      DIMENSION A(50,6,6),B(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 B(I,J) = A(N,I,J)
      RETURN
      END


      SUBROUTINE EQUAL(A,B)
      DIMENSION A(6,6),B(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 B(I,J) = A(I,J)
      RETURN
      END


      SUBROUTINE DEQUAL(A,B)
      DOUBLE PRECISION A,B
      DIMENSION A(6,6),B(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 B(I,J) = A(I,J)
      RETURN
      END
```

```
      SUBROUTINE CXMAT(A,B,C,N)
      COMPLEX A(N,N),B(N,N),C(N,N)
      DO 200 I = 1,N
      DO 200 J = 1,N
      C(I,J) = (0.,0.)
      DO 200 K = 1,N
  200 C(I,J) = C(I,J) + A(I,K) * B (K,J)
      RETURN
      END
```

```
      SUBROUTINE DEQUI(M,A,B)
      DOUBLE PRECISION A,B
      DIMENSION A(50,6,6),B(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 A(N,I,J) = B(I,J)
      RETURN
      END
```

```
      SUBROUTINE DADD(A,B,C)
      DOUBLE PRECISION A,B,C
      DIMENSION A(6,6),B(6,6),C(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 C(I,J) = A(I,J) + B(I,J)
      RETURN
      END
```

```
      SUBROUTINE DXMAT(A,B,C,N,L,M)
      DOUBLE PRECISION A,B,C
      DIMENSION A(L,M),B(N,N),C(N,N)
      DO 200 I = 1,N
      DO 200 J = 1,N
      C(I,J) = 0.
      DO 200 K = 1,N
  200 C(I,J) = C(I,J) + A(I,K) * B(K,J)
      RETURN
      END
```

```
C        SUBROUTINE DMINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     ..................................................................
C
      SUBROUTINE DMINV(A,N,D,L,M,ILNTH)
C     DIMENSION A(1),L(1),M(1)
      DIMENSION A(ILNTH),L(N), M(N)
C
C        ...............................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
      DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ...............................................................
C
C        SEARCH FOR LARGEST ELEMENT
C
```

```
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
```

```
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
```

```
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
C        SUBROUTINE MINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     ..................................................................
C
      SUBROUTINE MINV(A,N,D,L,M,ILNTH)
C     DIMENSION A(1),L(1),M(1)
      DIMENSION A(ILNTH),L(N), M(N)
C
C        ...............................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ...............................................................
C
C        SEARCH FOR LARGEST ELEMENT
C
```

```
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
```

```
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
```

```
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
      SUBROUTINE TIMES(AR,AI,BR,BI,CR,CI)
C     IMPLICIT REAL*8 (A-H,O-Z)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C        THIS ROUTINE PERFORMS MULTIPLICATION A * B = C IN COMPLEX MODE
C
      CR = AR * BR - AI * BI
      CI = AR * BI + AI * BR
C
      RETURN
      END
```

```
      SUBROUTINE DIVIDE(AR,AI,BR,BI,CR,CI)
C     IMPLICIT REAL*8 (A-H,O-Z)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C        THIS ROUTINE PERFORMS DIVISION A/B = C IN COMPLEX MODE.
C
C
      CR = (AR * BR + AI * BI) / (BR * BR + BI * BI)
C
      CI = (BR * AI - AR * BI) / (BR * BR + BI * BI)
      RETURN
      END
```

```
      SUBROUTINE COMPRD(A,AI,B,BI,R,RI,N,M,L)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C           SIMULATE COMPLEX MATRIX MULTIPLICATION WITH REAL ARRAYS
C
C
C
      DIMENSION A(36),B(36),R(36),AI(36),BI(36),RI(36)
      IR = 0
      IK = -M
      DO 10 K=1,L
      IK = IK + M
      DO 10 J=1,N
      IR = IR + 1
      JI = J - N
      IB = IK
      R (IR)= 0.
      RI(IR) = 0.
      DO 10 I = 1,M
      JI = JI + N
      IB = IB + 1
      CALL TIMES(A (JI),AI(JI),B (IB),BI(IB),DTEMPR,DTEMPI)
      R (IR) = R (IR) + DTEMPR
   10 RI(IR) = RI(IR) + DTEMPI
      RETURN
      END

      SUBROUTINE ADD(A,B,C)
      DIMENSION A(6,6),B(6,6),C(6,6)
      DO 200 I = 1,6
      DO 200 J = 1,6
  200 C(I,J) = A(I,J) + B(I,J)
      RETURN
      END

      SUBROUTINE XMAT2(A,B,C,N)
      DIMENSION A(N,N),B(N,N),C(N,N)
      DO 20 I=1,N
      DO 20 J=1,N
      C(I,J) = 0.
      DO 20 K=1,N
   20 C(I,J) = C(I,J) + A(I,K) * B(K,J)
      RETURN
      END
```

VARIANCES OF INDIVIDUAL POINTS

TOTAL VARIANCES = [illegible]
NUMBER OF POINTS = 40
SUMWT = [illegible]

STEP	AD.VARIANCE	VOL.VARIANCE	VERTICAL	
1	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]	[illegible]
7	[illegible]	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]	[illegible]
9	[illegible]	[illegible]	[illegible]	[illegible]
10	[illegible]	[illegible]	[illegible]	[illegible]
11	[illegible]	[illegible]	[illegible]	[illegible]
12	[illegible]	[illegible]	[illegible]	[illegible]
13	[illegible]	[illegible]	[illegible]	[illegible]
14	[illegible]	[illegible]	[illegible]	[illegible]
15	[illegible]	[illegible]	[illegible]	[illegible]
16	[illegible]	[illegible]	[illegible]	[illegible]
17	[illegible]	[illegible]	[illegible]	[illegible]
18	[illegible]	[illegible]	[illegible]	[illegible]
19	[illegible]	[illegible]	[illegible]	[illegible]
20	[illegible]	[illegible]	[illegible]	[illegible]
21	[illegible]	[illegible]	[illegible]	[illegible]
22	[illegible]	[illegible]	[illegible]	[illegible]
23	[illegible]	[illegible]	[illegible]	[illegible]
24	[illegible]	[illegible]	[illegible]	[illegible]
25	[illegible]	[illegible]	[illegible]	[illegible]
26	[illegible]	[illegible]	[illegible]	[illegible]
27	[illegible]	[illegible]	[illegible]	[illegible]
28	[illegible]	[illegible]	[illegible]	[illegible]
29	[illegible]	[illegible]	[illegible]	[illegible]
30	[illegible]	[illegible]	[illegible]	[illegible]
31	[illegible]	[illegible]	[illegible]	[illegible]
32	[illegible]	[illegible]	[illegible]	[illegible]
33	[illegible]	[illegible]	[illegible]	[illegible]
34	[illegible]	[illegible]	[illegible]	[illegible]
35	[illegible]	[illegible]	[illegible]	[illegible]
36	[illegible]	[illegible]	[illegible]	[illegible]
37	[illegible]	[illegible]	[illegible]	[illegible]
38	[illegible]	[illegible]	[illegible]	[illegible]
39	[illegible]	[illegible]	[illegible]	[illegible]
40	[illegible]	[illegible]	[illegible]	[illegible]

NORTH	EAST	TOTAL VARIANCE
.12503909E-05	.64339472E-01	.31298180E+01
.64749934E-01	.12047454E-05	.31300059E+01
.44819369E-04	.49760259E+00	.33373144E+01
.40769095E+00	.46245450E-04	.33374609E+01
.28798698E-03	.10203345E+01	.38168354E+01
.10134660E+01	.27102063E-03	.38170459E+01
.97754355E-03	.18059204E+01	.45216392E+01
.18072591E+01	.92168199E-03	.45220558E+01
.24696769E-02	.27772819E+01	.54111647E+01
.27804663E+01	.22347312E-02	.54120532E+01
.51690089E-02	.38832165E+01	.64502642E+01
.38829937E+01	.47249573E-02	.64519960E+01
.94929632E-02	.50945010E+01	.76087867E+01
.51165011E+01	.85772091E-02	.76127025E+01
.15964533E-01	.63857526E+01	.88601417E+01
.64046849E+01	.14183372E-01	.88659417E+01
.25061470E-01	.77539789E+01	.10182156E+02
.77532261E+01	.21891427E-01	.10190189E+02
.37294044E-01	.91214694E+01	.11555570E+02
.91536090E+01	.32321550E-01	.11566129E+02
.57152264E-01	.10531759E+02	.12964109E+02
.10590014E+02	.44304907E-01	.12977248E+02
.77136466E-01	.11951049E+02	.14393898E+02
.12028959E+02	.60589373E-01	.14409465E+02
.97438518E-01	.13360099E+02	.15835563E+02
.13474914E+02	.79557195E-01	.15859877E+02
.12702801E+00	.14773490E+02	.17273310E+02
.14399512E+02	.10205066E+00	.17290103E+02
.16211728E+00	.16157287E+02	.18704413E+02
.15314374E+02	.12796731E+00	.18720854E+02
.20308548E+00	.17510738E+02	.20120762E+02
.17735978E+02	.15752370E+00	.20135133E+02
.25034394E+00	.18837329E+02	.21516985E+02
.19068823E+02	.19079417E+00	.21527073E+02
.30423430E+00	.20123652E+02	.22889829E+02
.20398459E+02	.22781738E+00	.22891865E+02
.36526848E+00	.21368551E+02	.24232787E+02
.21695764E+02	.26836519E+00	.24219112E+02
.43336707E+00	.22571581E+02	.25545649E+02
.22938780E+02	.31285740E+00	.25518366E+02

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED

FINAL RESULTS

YRO DRIFT RATES, ALPHA -.[illegible]E-07 BETA .[illegible]E-09 GAMMA -.6290E-08

DEFLECTIONS OF VERT

XI	ETA	NORTH POS	EAST POS
.210000E-04	0.	0.	0.
.191201E-04	.153654E-05	0.	.250000E+06
.166245E-04	.961553E-06	0.	.500000E+06
.155531E-04	.112295E-05	0.	.750000E+06
.147065E-04	.961949E-06	.210500E+06	.875000E+06
.140719E-04	.867706E-06	.422010E+06	.100000E+07
.138850E-04	.821355E-06	.649500E+06	.112500E+07
.148065E-04	[illegible]	.111150E+07	.918490E+06
.102172E-04	[illegible]	.123650E+07	.591990E+06
.177352E-04	[illegible]	.131150E+07	.475480E+06
.210000E-04	0.	.114459E+07	.350480E+06

VARIANCE OF SOLUTION .2197E+01 SIGMA .2374E+00
NDIVIDUAL OBS ERRORS

1	.6411E-01
2	.1460E+01
3	.5991E-01
4	.4881E-01
5	.5040E-01
6	.3951E-01
7	.5194E-01
8	.5688E-01
9	.7981E-01
10	.6510E-01
11	.5910E-01
12	.4990E-01
13	.5066E-01
14	.5235E-01
15	.4758E-01
16	.5501E-01
17	.5741E-01
18	.5476E-01
19	.5058E-01
20	.5697E-01
21	.5660E-01
22	.5762E-01
23	.4137E-01
24	.5109E-01
25	.5606E-01
26	.5137E-01
27	.5115E-01
28	.6249E-01
29	.4735E-01
30	.4624E-01
31	.5051E-01
32	.5184E-01
33	.6009E-01
34	.5458E-01
35	.5440E-01
36	.5381E-01
37	.4915E-01
38	.4900E-01
39	.5357E-01
40	.5412E-01

UNCLASSIFIED UNCLASSIFIED UNCLASSI

VARIANCES OF INDIVIDUAL POINTS

TOTAL VARIANCE = [illegible]
NUMBER OF POINTS = 40
SUMMT = [illegible]

STEP	AO.VARIANCE	UO.VARIANCE	VERTICAL	
1	[illegible]	[illegible]	[illegible]	.12
2	[illegible]	[illegible]	[illegible]	.64
3	[illegible]	[illegible]	[illegible]	.44
4	[illegible]	[illegible]	.21794420E-08	.40
5	[illegible]	[illegible]	[illegible]	.28
6	[illegible]	[illegible]	[illegible]	.10
7	[illegible]	[illegible]	[illegible]	.97
8	[illegible]	[illegible]	[illegible]	.18
9	[illegible]	[illegible]	.27372394E-02	.24
10	[illegible]	[illegible]	[illegible]	.27
11	[illegible]	[illegible]	.17841250E-07	.51
12	[illegible]	[illegible]	[illegible]	.38
13	[illegible]	[illegible]	[illegible]	.94
14	[illegible]	[illegible]	[illegible]	.51
15	[illegible]	[illegible]	[illegible]	.15
16	[illegible]	[illegible]	.11261711E-04	.64
17	[illegible]	[illegible]	[illegible]	.25
18	[illegible]	[illegible]	[illegible]	.77
19	[illegible]	[illegible]	[illegible]	.37
20	[illegible]	[illegible]	[illegible]	.91
21	[illegible]	[illegible]	[illegible]	.53
22	[illegible]	[illegible]	[illegible]	.10
23	[illegible]	[illegible]	[illegible]	.73
24	[illegible]	[illegible]	[illegible]	.12
25	[illegible]	[illegible]	[illegible]	.97
26	[illegible]	[illegible]	[illegible]	.13
27	[illegible]	[illegible]	[illegible]	.12
28	[illegible]	[illegible]	[illegible]	.14
29	[illegible]	[illegible]	[illegible]	.18
30	[illegible]	[illegible]	[illegible]	.13
31	[illegible]	[illegible]	[illegible]	.21
32	[illegible]	[illegible]	[illegible]	.17
33	[illegible]	[illegible]	[illegible]	.27
34	[illegible]	[illegible]	[illegible]	.11
35	[illegible]	[illegible]	[illegible]	.30
36	[illegible]	[illegible]	[illegible]	.21
37	[illegible]	[illegible]	[illegible]	.38
38	[illegible]	[illegible]	[illegible]	.21
39	[illegible]	[illegible]	[illegible]	.43
40	[illegible]	[illegible]	[illegible]	.22

NORTH	EAST	TOTAL VARIANCE
.12503809E-15	.64339472E-01	.31298200E+01
.64345934E-01	.12047494E-05	.31300059E+01
.44319369E-04	.40760259E+00	.33373381E+01
.40729055E+00	.43245450E-04	.33374609E+01
.36358698E-03	.10030345E+01	.38169647E+01
.10034666E+01	.27102063E-03	.38170459E+01
.97754355E-03	.18059204E+01	.45220736E+01
.18072591E+01	.92138199E-03	.45220558E+01
.24696369E-02	.27772819E+01	.54122700E+01
.27804663E+01	.22947312E-02	.54120636E+01
.51650087E-02	.38832165E+01	.64526162E+01
.38895982E+01	.47248673E-02	.64519972E+01
.94929632E-02	.50945010E+01	.76131619E+01
.51065011E+01	.85772091E-02	.76127036E+01
.15964533E-01	.63353526E+01	.88678302E+01
.64046845E+01	.14183372E-01	.88659456E+01
.25061470E-01	.77339769E+01	.10194306E+02
.77632231E+01	.21891427E-01	.10190199E+02
.37294044E-01	.91214694E+01	.11574110E+02
.91636090E+01	.32021550E-01	.11566149E+02
.53152266E-01	.10531759E+02	.12991173E+02
.10590014E+02	.44364907E-01	.12977287E+02
.73106445E-01	.11951049E+02	.14432020E+02
.12028955E+02	.60680373E-01	.14409534E+02
.97436518E-01	.13369099E+02	.15837215E+02
.13474944E+02	.79867185E-01	.15858958E+02
.12772801E+00	.14773490E+02	.17342217E+02
.14699612E+02	.10205086E+00	.17290291E+02
.16210719E+00	.16157287E+02	.18794326E+02
.15314374E+02	.12796731E+00	.18721137E+02
.20316548E+00	.17513738E+02	.20235830E+02
.17705977E+02	.19752370E+00	.20135547E+02
.25034384E+00	.18837329E+02	.21661784E+02
.19068823E+02	.19079417E+00	.21527662E+02
.30423430E+00	.20127629E+02	.23068336E+02
.20398459E+02	.22761780E+00	.22892684E+02
.36525348E+00	.21368551E+02	.24452611E+02
.21685767E+02	.26836519E+00	.24220287E+02
.47336703E+00	.22570581E+02	.25812347E+02
.22336780E+02	.31285740E+00	.25519924E+02

VARIANCES OF INDIVIDUAL POINTS

TOTAL VARIANCES = [illegible]
NUMBER OF POINTS = [illegible]
SUMKT = [illegible]

STEP	AG.VARIANCE	VOL.VARIANCE	VERTICAL	HORI
1	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]	[illegible]
7	[illegible]	[illegible]	[illegible]	[illegible]
8	[illegible]	[illegible]	[illegible]	[illegible]
9	[illegible]	[illegible]	[illegible]	[illegible]
10	[illegible]	[illegible]	[illegible]	[illegible]
11	[illegible]	[illegible]	[illegible]	[illegible]
12	[illegible]	[illegible]	[illegible]	[illegible]
13	[illegible]	[illegible]	[illegible]	[illegible]
14	[illegible]	[illegible]	[illegible]	[illegible]
15	[illegible]	[illegible]	[illegible]	[illegible]
16	[illegible]	[illegible]	[illegible]	[illegible]
17	[illegible]	[illegible]	[illegible]	[illegible]
18	[illegible]	[illegible]	[illegible]	[illegible]
19	[illegible]	[illegible]	[illegible]	[illegible]
20	[illegible]	[illegible]	[illegible]	[illegible]
21	[illegible]	[illegible]	[illegible]	[illegible]
22	[illegible]	[illegible]	[illegible]	[illegible]
23	[illegible]	[illegible]	[illegible]	[illegible]
24	[illegible]	[illegible]	[illegible]	[illegible]
25	[illegible]	[illegible]	[illegible]	[illegible]
26	[illegible]	[illegible]	[illegible]	[illegible]
27	[illegible]	[illegible]	[illegible]	[illegible]
28	[illegible]	[illegible]	[illegible]	[illegible]
29	[illegible]	[illegible]	[illegible]	[illegible]
30	[illegible]	[illegible]	[illegible]	[illegible]
31	[illegible]	[illegible]	[illegible]	[illegible]
32	[illegible]	[illegible]	[illegible]	[illegible]
33	[illegible]	[illegible]	[illegible]	[illegible]
34	[illegible]	[illegible]	[illegible]	[illegible]
35	[illegible]	[illegible]	[illegible]	[illegible]
36	[illegible]	[illegible]	[illegible]	[illegible]
37	[illegible]	[illegible]	[illegible]	[illegible]
38	[illegible]	[illegible]	[illegible]	[illegible]
39	[illegible]	[illegible]	[illegible]	[illegible]
40	[illegible]	[illegible]	[illegible]	[illegible]

VERTICAL	NORTH	EAST	TOTAL VARIANCE
.53797037E-05	.19780893E-05	.10178355E+00	.31672648E+01
.11930652E-10	.10179377E+00	.19058792E-05	.31674544E+01
.62400017E-04	.71833172E-04	.65978015E+00	.35894627E+01
.21754442E-08	.65984220E+00	.69310086E-04	.35896386E+01
.83247736E-03	.46154587E-03	.16514413E+01	.44655495E+01
.35046936E-07	.16521511E+01	.44112063E-03	.44659012E+01
.10952727E-02	.16131943E-02	.30211615E+01	.57379507E+01
.23010371E-06	.30234026E+01	.15215627E-02	.57387991E+01
.27372394E-02	.41291193E-02	.47177894E+01	.73544369E+01
.94969592E-06	.47232106E+01	.38376583E-02	.73563508E+01
.97241250E-02	.87445960E-02	.66946980E+01	.92675773E+01
.29559162E-05	.67057405E+01	.80024768E-02	.92714171E+01
.10438953E-01	.15267543E-01	.89097296E+01	.11435165E+02
.59841995E-05	.89308186E+01	.14707150E-01	.11443149E+02
.17810344E-01	.27682029E-01	.11323657E+02	.13817713E+02
.11281711E-04	.11358154E+02	.24612070E-01	.13829844E+02
.26235961E-01	.43957995E-01	.13901671E+02	.16360896E+02
.26042041E-04	.13954644E+02	.38434787E-01	.16398163E+02
.42461379E-01	.66151831E-01	.16611913E+02	.19093412E+02
.52647846E-04	.16689369E+02	.56865985E-01	.19116753E+02
.61131695E-01	.95319070E-01	.19425821E+02	.21927401E+02
.96869278E-04	.19533443E+02	.80569532E-01	.21957415E+02
.64976523E-01	.17256852E+00	.22317441E+02	.24857843E+02
.16608876E-03	.22464737E+02	.11017047E+00	.24894806E+02
.11388721E+00	.17845839E+00	.25266194E+02	.27865330E+02
.21867321E-03	.25468927E+02	.14657936E+00	.27919553E+02
.15017766E+00	.23504525E+00	.28246354E+02	.30923098E+02
.42963438E-03	.28491210E+02	.18925453E+00	.30969193E+02
.19352818E+00	.30297267E+00	.31241679E+02	.34019583E+02
.67901044E-03	.31550892E+02	.23977243E+00	.34069459E+02
.24431439E+00	.38729631E+00	.34235688E+02	.37137991E+02
.92243459E-03	.34618934E+02	.29314510E+00	.37189127E+02
.30489235E+00	.47704138E+00	.37213803E+02	.40264955E+02
.12975288E-02	.37681083E+02	.36470476E+00	.40313833E+02
.37409462E+00	.58519625E+00	.40163200E+02	.43388669E+02
.17842784E-02	.40724740E+02	.43971725E+00	.43430864E+02
.45349117E+00	.70907441E+00	.43071464E+02	.46499341E+02
.24951300E-02	.43727616E+02	.52290358E+00	.46516678E+02
.54295150E+00	.84888835E+00	.45931182E+02	.49583468E+02
.32845854E-02	.46701589E+02	.61529966E+00	.49585276E+02

VARIANCE OF SOLUTION .1160E+01 SIGMA .1732E+00

NDIVIDUAL RSS ERRORS

1	.2449E-02
2	.1301E+00
3	.4506E-01
4	.2397E-01
5	.6209E-01
6	.5968E-02
7	.7009E-01
8	.1876E-01
9	.1958E-01
10	.1017E-01
11	.2910E-01
12	.1995E-02
13	.3056E-01
14	.7513E-04
15	.2325E-01
16	.4059E-03
17	.1838E-01
18	.1422E-02
19	.1216E-01
20	.5865E-02
21	.1953E-01
22	.1206E-01
23	.1825E-01
24	.1738E-01
25	.1567E-01
26	.2327E-01
27	.1708E-01
28	.3468E-01
29	.1717E-01
30	.4359E-01
31	.1607E-01
32	.5037E-01
33	.1556E-01
34	.6509E-01
35	.1712E-01
36	.7963E-01
37	.1745E-01
38	.9073E-01
39	.1911E-01
40	.1147E+00

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED

RESULTS WITHOUT VERTICAL GYRO DRIFT

GYRO DRIFT RATES. BETA .4973E-09 GAMMA -.3002E-09

DEFLECTIONS OF VERT

XI	ETA	NORTH POS	EAST POS
.250000E-04	0.	0.	0.
.217271E-04	.155619E-05	0.	.250000E+06
.211841E-04	.103955E-05	0.	.500000E+06
.212694E-04	.103512E-05	0.	.750000E+06
.210354E-04	.799941E-06	.216500E+06	.875000E+06
.209314E-04	.709922E-06	.433010E+06	.100000E+07
.207578E-04	.622617E-06	.649500E+06	.112500E+07
.205745E-04	.378144E-06	.111151E+07	.956430E+06
.203271E-04	.265447E-06	.123650E+07	.591990E+06
.201513E-04	.110776E-06	.136150E+07	.475480E+06
.210010E-04	0.	.114499E+07	.750480E+06

VARIANCE OF SOLUTION .2989E+01 SIGMA .8724E-01
NDIVIDUALSSQERRORS

1	.1094E-07
2	.1072E+00
3	.5790E-02
4	.2379E-02
5	.8917E-02
6	.7130E-03
7	.4669E-02
8	.4087E-02
9	.7109E-02
10	.1370E-01
11	.4133E-02
12	.4610E-02
13	.6241E-02
14	.1075E-02
15	.5165E-02
16	.3866E-02
17	.3639E-02
18	.6453E-02
19	.4518E-02
20	.5129E-02
21	.5291E-02
22	.3814E-02
23	.5379E-02
24	.3175E-02
25	.3571E-02
26	.5903E-02
27	.4941E-02
28	.4727E-02
29	.5902E-02
30	.3356E-02
31	.5577E-02
32	.2448E-02
33	.3305E-02
34	.5095E-02
35	.3489E-02
36	.1206E-01
37	.5925E-02
38	.4191E-02
39	.9270E-02
40	.5043E-02

RESULTS WITHOUT GYRO DRIFTS

DEFLECTIONS OF VERT

XI	ETA	NORTH POS	EAST POS
.200000E-04	0.	0.	0.
.208162E-04	.250741E-05	0.	.250000E+06
.206150E-04	.141790E-05	0.	.500000E+06
.206195E-04	.141290E-05	0.	.750000E+06
.205760E-04	.646298E-06	.210500E+06	.875000E+06
.205264E-04	.532639E-06	.437010E+06	.100000E+07
.204300E-04	.251687E-06	.649500E+06	.112500E+07
.206185E-04	-.200503E-06	.111150E+07	.908490E+06
.201387E-04	-.567275E-06	.123650E+07	.691990E+06
.200726E-04	-.117944E-05	.130150E+07	.475480E+06
.200000E-04	0.	.114499E+07	.353480E+06